EIM CONTROLS

User's Guide

AB1746-C Network Master (Version 5.21)

For
Allen-Bradley Series SLC-500 Programmable Controllers

(Date: 2004-11-18)

User's Guide

Controlinc Network Master … AB1746-C (Version 5.21)

For
Allen-Bradley Series SLC-500 Programmable Controllers

Table of Contents

0. Quick Start Information

1. As a minimum, you should have access to the following reference documents:

    - **This manual (1746-C User's Guide)**
    - **Allen-Bradley SLC-500 BASIC User's Manual (1746-BAS & 1746-BAS-T)**
      Publication No. 1746-UM004A-US-P 2000
    - **EIM Controlinc 320A Quick Startup Guide, Rev. F or later**
    - **EIM Controlinc 320B Quick Startup Guide, Rev. A or later**

2. Refer to Section 2 of this manual on how the jumpers should be set on the BASIC module.

3. The 1746-C module comes from EIM pre-programmed with the network master program in it. However, if for some reason the module needs to be flashed again (ex: a field upgrade) then refer to Section 5 on installing the firmware in the module.

    Note: Remember, you will need to connect a serial cable between the 1746-C and the computer. The correct cable is a **Null Modem Cable** connected between the PRT1 port (top DB9 connector on the module) and the serial communications port on the computer used for the upload.

4. Ensure the PLC rack power is turned off.
   Install the module in the correct slot in the PLC rack. Connect all cables and power the system up.

5. The PLC-RLL will start interfacing with the 1746-C after the 1746-C performs initial scans of the network on power up.

NOTES: To ensure optimal performance, some things the PLC should do …

1. Ensure table [0] is set correctly and repeated in a timely manner. Allow for frequent table [0] read backs.

2. Restrict the rate of data writes to the actuators … only write as fast as required to adequately control the valve.
   (*cuts down on the interruptions to scanning the network … gathering data*)

3. Expedite responses to M1 transfer requests made by the 1746-C. Delaying them can slow 1746-C operations.

4. Only activate writing to or reading from tables if actually in use (or when needed).
   (*ex: do not write to the analog output table if analog outputs are not used*)
   (*ex: do not read totalizer data if its not being used*)
   (*ex: only poll for "static" [non-changing] data one time and stop*)

   (*prevents unnecessary delays to gathering the more important data from the network*)

Remember: The 1746-C uses explicit read/write commanding of the tables such that if word [1] in any table is zero (0) then it is a table read request by the PLC. Otherwise, the command is a table write command.

1. Introduction

1.1. Overview of 1746-C System

This document is intended for end users as a Guide in applying an EIM AB 1746-C Network Master.

The software that enables network operation is called a *communication driver.* It is used in conjunction with an Allen-Bradley 1746-BAS module thereby creating the 1746-C network master module. The 1746-C allows an AB SLC-500 controller to acquire data from and send information to an EIM Controlinc Actuator Network.

GENERAL SYSTEM PURPOSE:

The EIM 1746-C network master serves as a data concentrator for applications that use an AB SLC-500 Programmable Logic Controller (PLC). The 1746-C module must be located in a slot in the local chassis (it cannot be mounted in a chassis that is remote from the SLC CPU).

In general, the network master off-loads network communication and management tasks from the actual controlling equipment. The main function of the 1746-C is to provide the interface between a PLC and its network of valve actuators. Acting as a Host, the PLC controls the network by sending data to the 1746-C for routing to a particular actuator. The 1746-C network master polls the individual actuators for information and makes it available to the PLC when requested.

A typical ring topology network and network master are illustrated in Figure 1. The 1746-C module serves as a master within this Modbus (Modbus RTU) master/slave network. The module will manage network operation by keeping an orderly cycle of data transfers between itself and each slave (valve actuator).

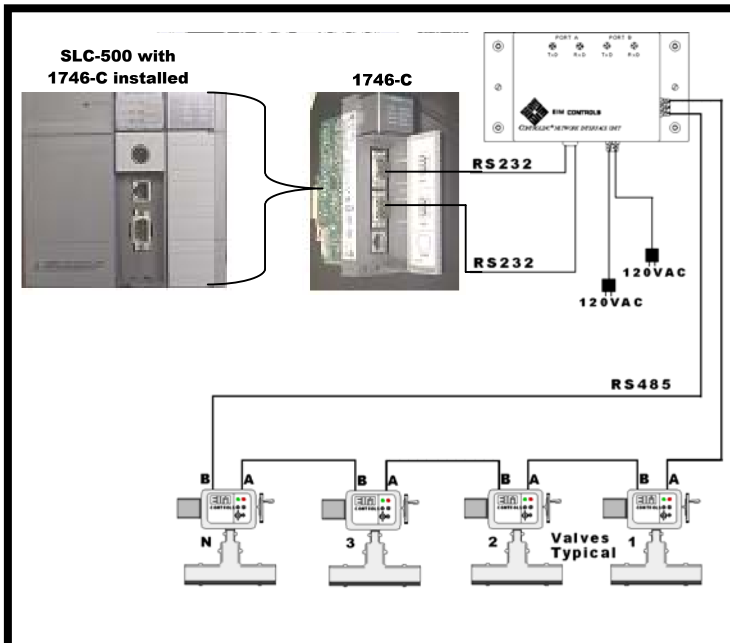


Figure 1

The 1746-C master will handle network communication, error detection, alarming, and network recovery.
Other features include:

- Reporting of inaccessible actuators
- Reporting of network faults
- Emergency shutdown broadcasting
- Minimal Interfacing with the Relay Ladder Logic (RLL) program in the PLC CPU module

Each 1746-C module can support a single network of up to 60 valve actuators. Multiple modules can be installed in an SLC-500 system rack to provide support for multiple networks (up to 60 actuators each).

Note 1: *It is important to note that the 1746-C Network Master may be used in any slot of the 1746 rack, not just slot 1 next to the SLC-500 CPU.*

1.2. Overview of Changes in Version 5.21

For the 1746-C system, there is no previous version of the product to reflect functional changes to.

However, any later releases of this document for this version of the product (5.21) are to only correct entries in this document … primarily grammatical or "typos" that are discovered in the future. Major ideas will be documented with a separate "line entry".

Major document changes:

1. This document: 2004-11-18
   Previous document: 2004-08-17
   Major Change: Memory Maps adjusted.

2. New document: 2004-08-17
   Previous document: 2004-08-06
   Major Changes: Memory Maps adjusted and this section added.

1.3. Reference Manuals

- **Allen-Bradley SLC-500 BASIC User's Manual (1746-BAS & 1746-BAS-T)**
  Publication No. 1746-UM004A-US-P 2000

  This manual is required for installing the module and for proper uploading of this driver into the hardware.

- **EIM … Controlinc 320A Quick Startup Guide, Rev. F or later**
  Publication No. ECL-4004-0102

  This manual is used for specific information on the Controlinc 320A Controller card located in the valve actuator. This includes networking, setup and available options.

- **EIM … Controlinc 320B Quick Startup Guide**
  Publication No. ECL-4005-0404

  This manual is used for specific information on the Controlinc 320B Controller card located in the valve actuator. This includes networking, setup and available options.

- **EIM … TEC2000 "Document … TBD"**
  Publication No. ?????????

  At the time of this printing, this TEC2000 publication had not been made available.

- Other Allen-Bradley manuals specific to the SLC-500 being applied. These may be required to implement the necessary Relay Ladder Logic (RLL) for application of the driver. For instance, this may include:

  - **Allen Bradley SLC-500 Instruction Set Reference,**
    Publication 1747-RM001D-EN-P (November-2003)

    This reference includes information in Appendix E on the M0 & M1 File Data Transfer handling.

1.4. EIM Equipment Compatibility

The 1746-C is compatible with the following EIM equipment:

- 320A … version 1.17 and later.
- 320B … all
- TEC2000 … all

Other points:

- The 1746-C derives the system type of each actuator during the diagnostic scan and stores the results in table [18].
- The 1746-C derives the firmware Version ID for the 320A & 320B systems during the diagnostic scan and stores the results in table [19].
- Since the TEC2000 system has several micro-controllers with firmware, a special request must be made to them to derive the firmware version ID for each controller in the unit.
- The SSR bit is only checked and set on 320A actuators.

Note: *For maximum compatibility with the TEC2000 systems, it is recommended that you enable monitoring the TEC2000 Status Inputs (tables [21-24]). This way, you can ensure you have all critical status information for any TEC2000 system on the network. For more information, refer to the section on Tables [21-24].*

Note: *This document references the addressable memory ranges ("Memory Maps") of several EIM actuators (ex: M2CP–320B, TEC2000). This is only for the reader's convenience. The specific details in the memory maps are only correct as of the date this manual was published. Therefore, to ensure you have the most current memory information, please refer to the technical information for that particular actuator.*

1.5. Firmware

The firmware is a "driver program" which is a compiled application program (not an interpreted one). It is loaded and stored in the module's 32K EEPROM (A-B P/N 1747-M2). The user is required to provide configuration information from the PLC Relay Ladder Logic (RLL) for each specific application (the RLL must load table [0] with appropriate configuration information).

2. <u>1746-C Hardware</u>

2.1. <u>Hardware Setup</u>

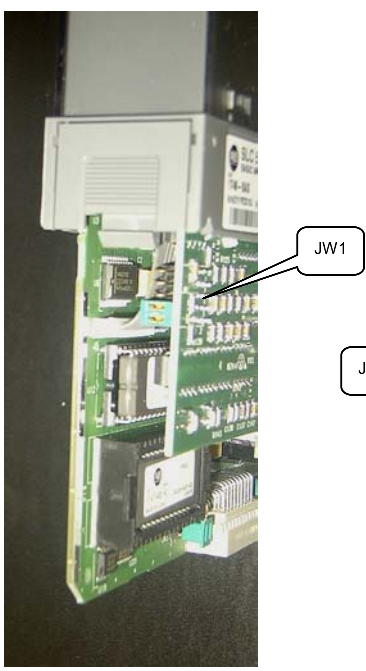
Refer to Chapter 1 of the 1746-BAS manual. The following hardware jumper settings are required. If the 1746-BAS module was supplied by EIM, jumpers are already set and no other settings are required.

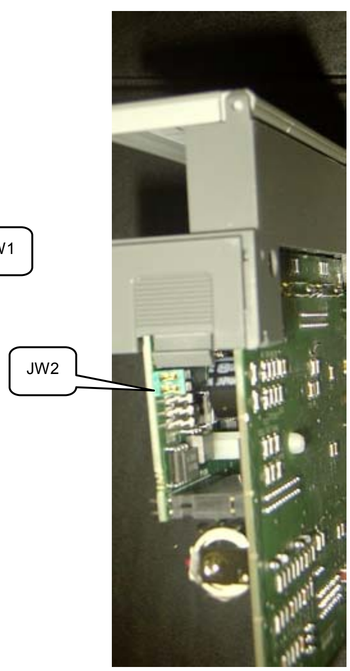
For the 1746-C to run, the jumper settings should be set as follows:

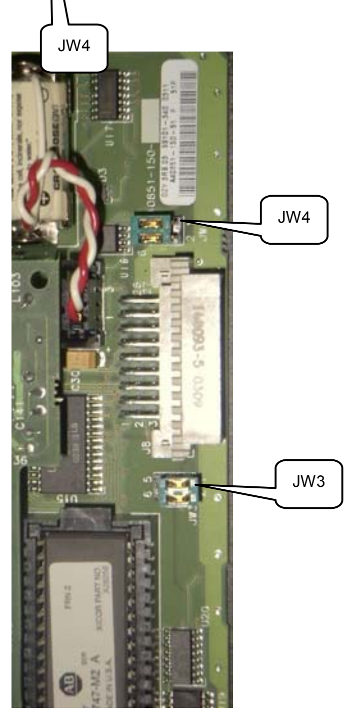
- JW1 - enable RS232
- JW2 - enable RS232
- JW3 - M2 EEPROM

- JW4 - PRT1 = ASCII
  - PRT2 = ASCII
  - DH485 = PGM

This jumper configuration is normally performed by EIM before shipping the 1746-C.







2.2. LED Utilization

2.2.1. Green (PRT1 & PRT2) and Yellow (LED1 & LED2) LED Usage

There are 2 “Yellow” LED lights on the front of the 1746-C module

- LED1.
- LED2.

There are currently 3 general functions operating these LEDs:

1. After CPU restart, both LED1 & LED2 are ON until the 1746-C finishes initialization and starts polling the network.

2. After initialization, LED1 is used as a “starting a new scan” indicator by toggling on/off at the start of each scan. Ex: Starting scan 1: LED1 is ON.
   Starting scan 2: LED1 toggles to the OFF state.
   Starting scan 3: LED1 toggles to the ON state.

3. After initialization, LED2 is used as an “entire network is down” indicator. If the 1746-C ever detects zero (0) actuators on the network, it will turn LED2 on and immediately enter a diagnostic scan and remain there until an actuator is found to be on-line and communicating.

There are 2 communication LED lights (green LED lights) on the 1746-C module:

- PRT1: Port 1 Transmit … lights up when transmissions are going out port 1.
- PRT2: Port 2 Transmit … lights up when transmissions are going out port 2.

PRT1 & PRT2 (Green) LED App Note: Green LED Lights & System Initialization with Table [0]

After startup, if table [0] is not initialized shortly after the 1746-C finishes its initial diagnostic scan, the program starts running with factory default settings. This means that unless the network actually has 60 actuators on it, the 1746-C will believe the "other valves" are just off line.
Plus, an operator can often determine when the PLC-RLL actually writes to table [0] (configures the system) by watching the green LED lights.

For instance …

- On a network that actually has 25 valves on it (addressed 1-25), until table [0] – word [4] gets initialized with 25, the program thinks that there are still supposed to be 60 actuators on the network … only the last 35 happen to be "off line".

- Then (if viewing the green LED lights) when a scan starts (LED1 changes state), you will see the green LEDs "flash" while the 1746-C communicates with valves 1-25.

  Then you will observe a "dead space" in time when the program is trying to determine if any of the final 35 valves "it believes to be connected" are available (*and of course none are because the network only has 25 valves on it*).

- As soon as a new scan starts (all "60" valves have been checked and ready to scan the network again … LED1 changes state) then the green LEDs will start flickering again while the 1746-C communicates with valves 1-25.

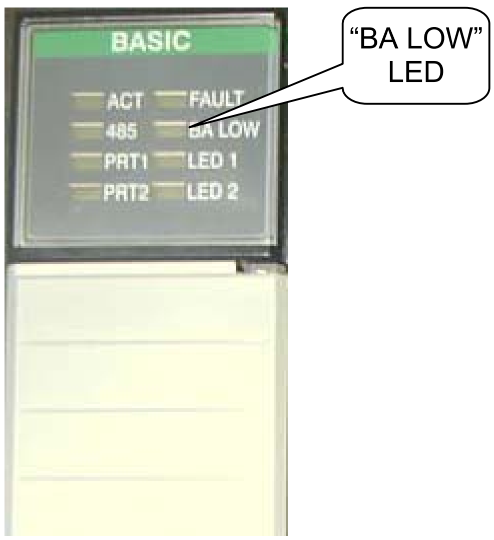
  Then the "dead space" time will be observed again.

  This sequence will repeat until the PLC program initializes the 1746-C by writing to table [0].

2.2.2. Red "BA LOW" LED Usage

The red "BA LOW" LED indicates low battery status. The purpose of the battery in the BASIC module is to back up portions of RAM and other resources. For guaranteed long-term repeatability of operations, the EIM 1746-C program does not utilize any of these resources. Therefore, it doesn't matter what state the battery is in (installed and fully charged, inline and dead or totally removed from the unit), the EIM network master program will always run the same.

This also means that a new 1746-C module might run with the BA LOW LED off for a while and then turn on when the battery dies. This is normal and has NO effect on the system. However, if for some reason you want to replace the battery, you can order it directly from Allen-Bradley.



3. Network Setup and Connecting to the NIU

In general, the EIM NIU ("Network Interface Unit") is a "beefed up" and configurable RS232 ←→ RS485 converter.

The ports on the 1746-C, Port1 (PRT1) and Port 2 (PRT2), are configured for serial communication using RS-232.

The network communication parameters are pre-configured and fixed at 9600 baud – 8 bit – no parity – 1 stop bit.

The NIU isolates and protects the 1746-C and the PLC from the network and is powered by its own 120 AC circuit. Its primary function is to convert RS232 communications to RS485 communications. The RS485 connection then communicates to a ring or ring network.

Note: To help with isolation, there are two (2) entirely separate boards in the NIU. However, because of this, there are two (2) 120 VAC connections to the NIU. Ensure that both are connected (you can jumper the circuits together).

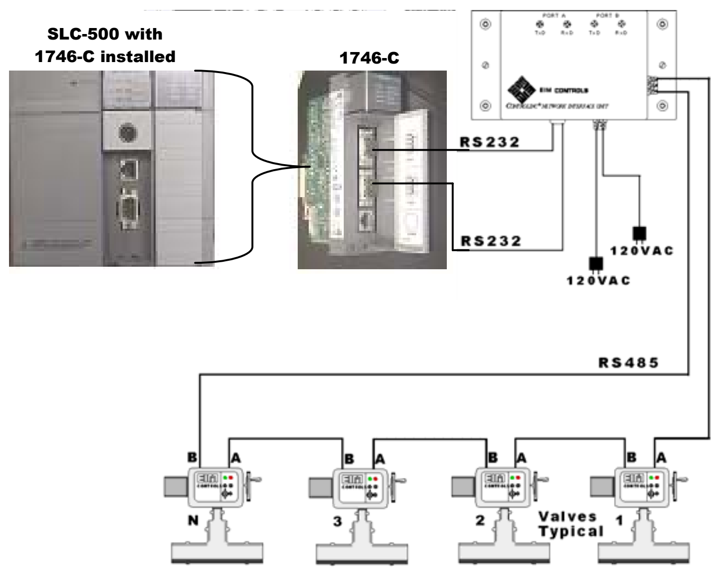


Figure 1

The Controlinc network is connected to the 1746-C module via the Network Interface Unit (NIU). It doesn't matter if you connect:

- Port 2 (PRT2) (bottom DB9 connector): connects to the NIU at the port labeled "HOST A".
- Port 1 (PRT1) (top DB9 connector): connects to the NIU at the port labeled "HOST B".

or

- Port 1 (PRT1) (top DB9 connector): connects to the NIU at the port labeled "HOST A".
- Port 2 (PRT2) (bottom DB9 connector): connects to the NIU at the port labeled "HOST B".

However, it is recommended that you remain consistent.

NOTE: You can always connect a PC to the network at the NIU if you …

- Disconnect the cable connecting the 1746-C and NIU.
- Connect the PC using a (serial cable + NULL modem) to the DB9 connector on the NIU.

A typical E>Net network connection format is in a ring …

- It starts from Port 1 (PRT1) (the top DB9 connector) on the 1746-C as RS232 and connects to the NIU at the port labeled “Host A”.
- It exits the NIU at the port labeled “Port A” as an RS485 circuit and proceeds to the first valve actuator, normally addressed as #1, in port A.
- It exits the actuator from Port B and proceeds to the port A next actuator (address #2) and on until the last actuator on the network is connected.
- The RS485 network then exits port B on the last actuator returns to the NIU at the port labeled “Port B”.
- The network connection is then transformed back to an RS232 format and exits the NIU via the port labeled “Host B”
- It connects to the 1746-C via port2 (PRT2) (the bottom DB9 connector).

3.1. Field Connections at the Actuator

Communication connections and wiring are important for the network and the master.

(*The following information was primarily derived from the* ***Controlinc Quick Startup Guide***)

3.1.1. Step 1. Plan the Network Topology

Before connecting actuators, the entire network layout should be planned. Topologies may be bus, redundant bus, E>Net, redundant E>Net, E>Net ring, and redundant E>Net rings. Planning should include node addressing, wire routing, terminations, and grounding.

3.1.2. Step 2. Select Network Cable

Ensure correct cable is being used.

- Networks require twisted pair and shielded cable with characteristic impedance between 50 and 120 Ohms.
- Capacitance between conductors must be less that 30 pF/Ft (98 pF/M); 10-15pF/Ft is ideal.
- Shielding maybe aluminum foil with drain wire.
- If cable has multiple pairs, then individual pair shielding is required.
- Only cables with stranded conductors are recommended.
- Insulating and outer jacket materials must be selected for the application environment.

The following are acceptable Belden or equivalent cables for most network applications.

AWG	20	18	16	14
Beldon #	8762	8760	8719	8720
Rating	12.8 Pf/fT	12.8 Pf/fT	12.8 Pf/fT	12.8 Pf/fT

3.1.3. Step 3. Route Cable away from Electrical Interference

Network cables should enter the electrical enclosures and the bottom or lowest point (on 320A systems, this is near the transformer end and normally in a counter clockwise direction to the topside of the TBM). Never install network cable in the same conduit with power conductors. Never route the network cable through the high voltage contactor area. On 320A systems, the cable should never lie across the TBM or hinder the protective cover of the TBM. Always use the shortest distance and keep access cable to a minimum.
See Figure 3-1

Figure 3-1

Correct termination of the Network to a Controlinc 320A Actuator

3.1.4. Step 4. Observe Polarity and Network Grounding

Each network connection is polarized + and - on wiring diagrams. Always use consistency in wiring and the use of wire colors to track polarity. The cable shield (or "drain wire") must be connected to the designated "shield" terminal at each port of each actuator. The shield must be connected to earth ground at only one point. Some networks require a jumper between the shield connections on ports A & B of the actuator to carry the shielding through the network. The shield connection of each actuator is isolated from earth ground.
Do not allow the shield to touch other circuits or the metal enclosure.

3.1.5. Step 5. Wire Preparation and Connections

Screw terminal connections on the TBM and in the TEC2000 terminal chamber have wire clamps, which will accept conductors with out terminals. Wire terminals may be applied if desired but are not required. Strip conductor insulation back 3/8" when connecting directly to the TBM screw terminals. Do not allow wire clippings to fall on the TBM or into the enclosure.
Protect the conductors and shield to prevent them from contacting any other circuits or earth ground. Use plastic electrical tape or heat shrink tubing to prevent bare conductors from contacting other circuits or earth ground. *See Figure 3-1*

3.1.6. Step 6. Test Network

Use EIM's Configuration and Control Utility (CCU) to test the network prior to connecting to the host or network master. The CCU is a Windows application, which will run on a laptop. Use a RS232 to RS485 adapter or EIM's Network Interface Unit (NIU) to connect the laptop to the network. Test each actuator, one at a time, to determine that all network connections are good and each actuator is functional via the network in remote.

3.2. Field Network Cable Connection to the NIU

Refer to Figure 3-2 for terminals on the EIM Network Interface Unit (NIU). The NIU is a dual port RS232 to RS485 converter specifically designed for ring networks.

Connect the twisted shielded pair cable of the field network to NIU Terminals 1 (-), 2 (Shield), 3 (+) of J1 of Port A.

The network shield must be connected to a good earth ground at only one shield terminal point on the NIU.

The network cabling should use recommended cabling for an RS-485 network able to handle the distance desired.

There should be little, if any, electrical noise on the network for the entire length of the network.

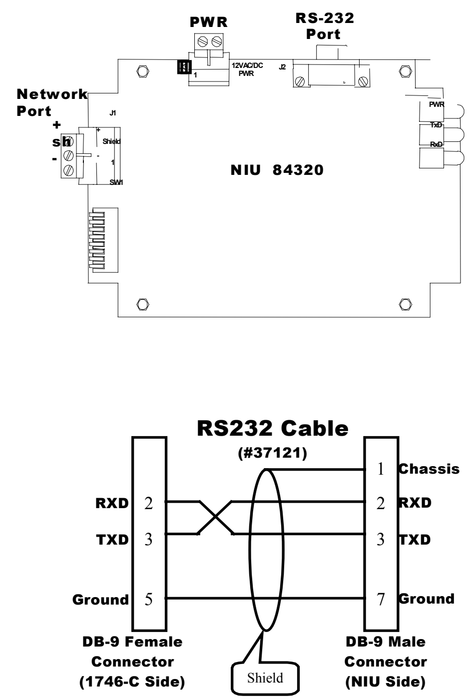


Figure 3-2

3.3. Cable Connection Between the NIU and the Network Master

Refer to the "**Allen-Bradley SLC-500 BASIC User's Manual (1746-BAS & 1746-BAS-T) Publication No. 1746-UM004A-US-P 2000"** for additional information on the pin-out of the male DB9 connections used for network ports one and two.

Jumpers JW1 & JW2 must be set for RS-232 when connecting with EIM's Network Interface Unit (NIU) P/N 84320 or a user-supplied modem.

Refer to Figure 3-2 for the pin-out of the DB-9 female connector.

4. 1746-C General Theory of Operation

4.1. Operational Overview

The interface for the PLC with the actuators (through the 1746-C module) is with an array of 25 tables … 64 words per table with the table number (command) in Word # [0] of each. This is based on the M0/M1 interface protocol established by Allen-Bradley for the SLC-500 systems.

The 1746-C operates using a "Non-Interruptible Time-Slice Process Allocation" communication methodology. This means that it must finish executing the process that is executing in its allocated time-slice before another process can operate. The two processes being scheduled are:

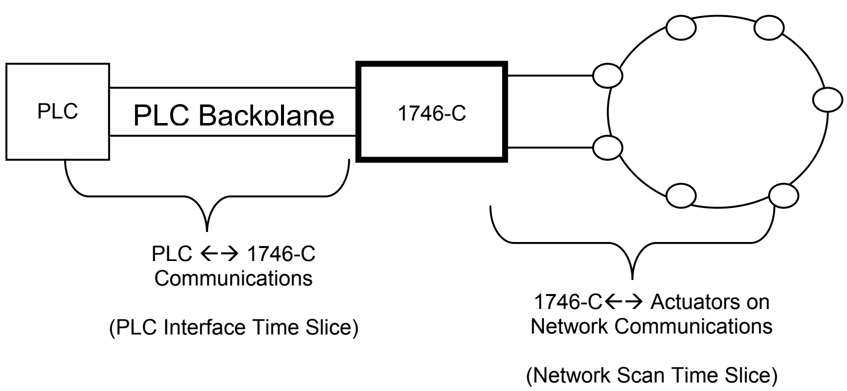
- Scanning the network for data.
- Interfacing with the PLC.

This means that the 1746-C will ignore any new commands from the PLC while scanning the network (during the network scan time-slice) and no network scanning will take place while processing a command from the PLC (during the PLC interface time-slice).

4.1.1. Diagram Describing Relationship Between Process Time Slices

Process time slicing is how the 1746-C divides its time when performing its 2 primary communication tasks … communicating with the PLC and communicating with the actuator network.

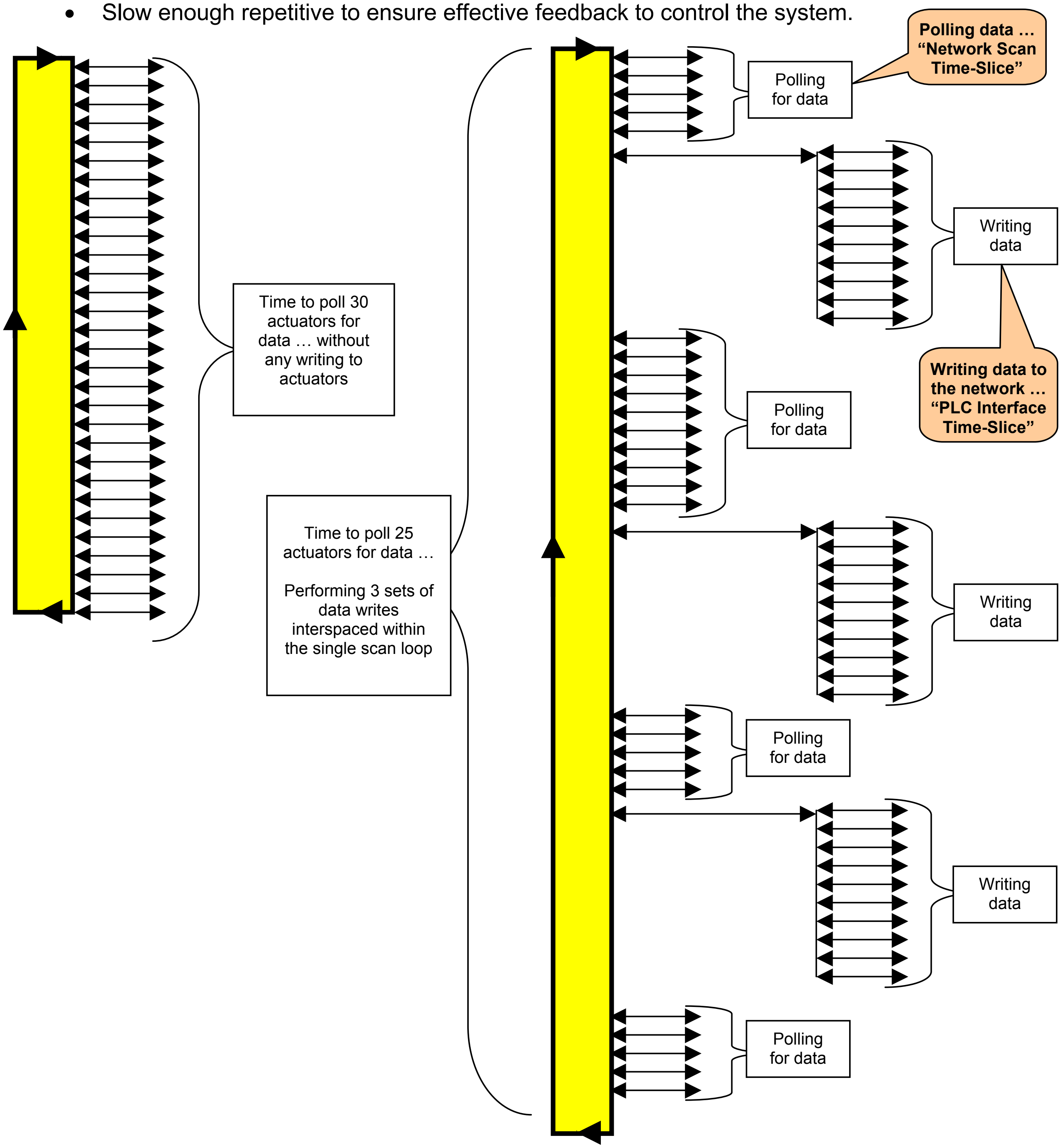
This diagram helps demonstrate the relationship between the two communication processes.



4.1.2. Diagram Describing Network Scanning & Actuator Writing Logic

This example demonstrates how a normal scan cycle gets elongated with data writes to the actuators. The more writes being performed the longer it will take to complete data scans. This is where the PLC/HMI must make a value decision for the best fit for their application:

- Fast enough repetitive writes to control the system
- Slow enough repetitive to ensure effective feedback to control the system.



4.1.3. Diagram Demonstrating Adding Additional Valves in Valve Scan Time-Slice

This example demonstrates how increasing the additional number of valves to be scanned before servicing the PLC communication task decreases the total network scan time.

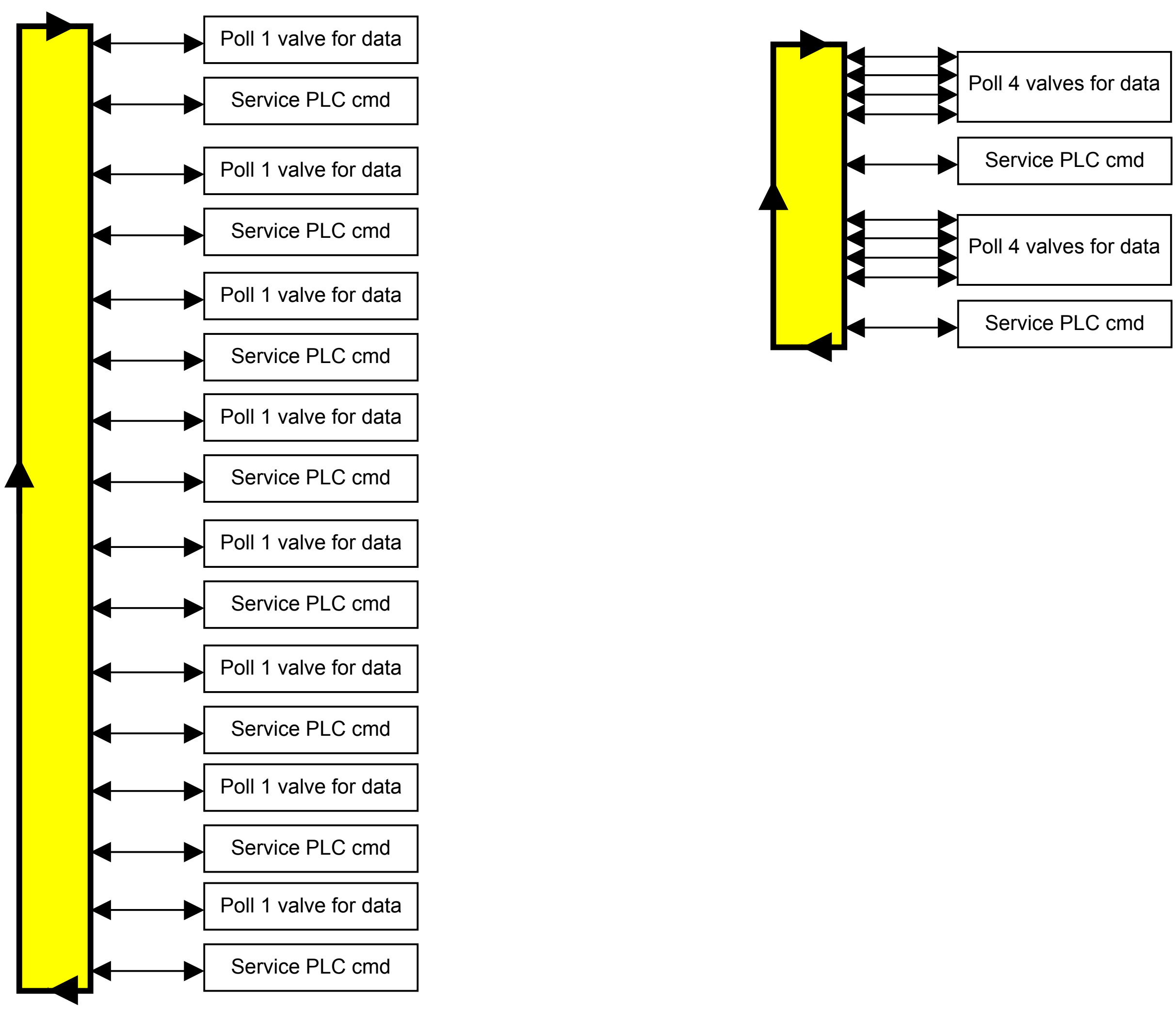
This is where the PLC/HMI must make a value decision for the best fit for their application:

- Increasing the number: Speeds up data collection Lowers PLC control capabilities.
- Decreasing the number: Slows down data collection Increases PLC control capabilities.

Example of scale for time to scan network … different values for Additional valves in scan time slice:

Network size: 8
Additional valves: 0

Network size: 8
Additional valves: 3

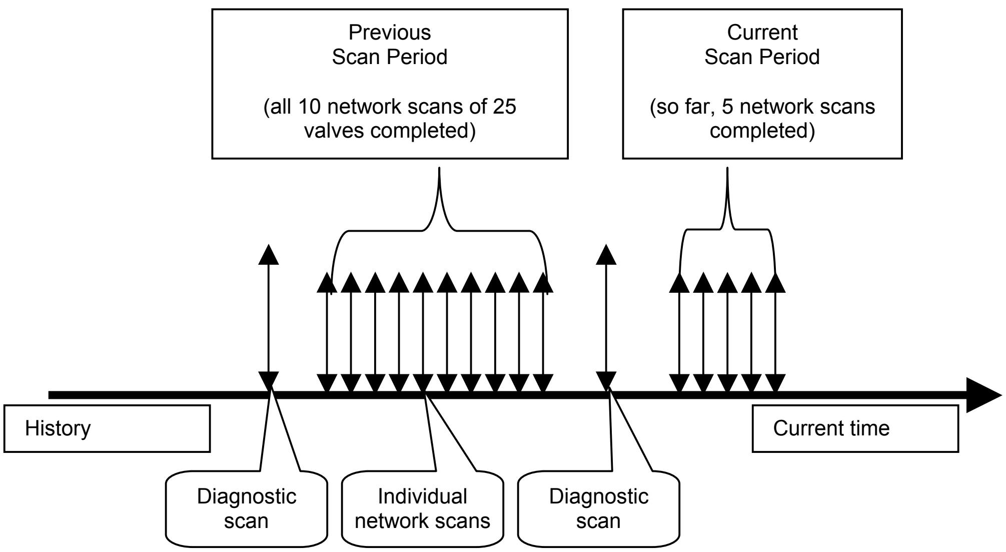


(Note the relative time differences are for illustration purposes only … not to any exact scale.)

4.1.4. Diagram Describing Relationship Between Scan Period & Diagnostic Scan

This example demonstrates the relationship between the diagnostic scans, scan periods, the current scan period, the previous scan period.

This example: a scan period value of 10 and 25 valves on the network. This means that all the valves on the network will be scanned 10 times before the scan period is over … and before the next diagnostic scan is performed.



4.1.5. 1746-C System Operations at Power Up

The 1746-C software boots up on a power up of the 1746-C module. The general sequence is:

1. Illuminate LED1 & LED2 as an indication that the software is initializing itself and the network.
2. Perform basic internal initializations.
3. Perform the initial diagnostic scan of the network and identify all actuators on the network.
4. Finish internal initializations.
5. Turn off LED1 & LED2.
6. Start standard scanning of the network. Initialization is finished. Normal operations (scanning) have commenced and are allowing PLC communications.
   (LED1 toggles ON during the start of this first scan)

Note 1: *Remember that the PLC cannot perform any block transfers until after the network master has finished polling the network for status information and LED2 has been turned off and the 1746-C starts indicating that it is ready for data (after scanning starts).*

Note 2: *Even though LED1 turns off immediately after the initial scan after power up and then toggles back on after normal operations start, the user will not normally see this … at the most it would be a small flicker. Therefore, to the user that is observing the LED behavior after restart on a system that has actuators communicating, it will appear that …*

During the initial scan when the PLC cannot talk to the 1746-C …
LED1 == ON
LED2 == ON.

Immediately after the initial scan when normal operations start (start of 1st normal scan):
LED1 == ON
LED2 == OFF.

4.2. Network Interface (Scan) Time-Slice (1746-C ↔ Network Communication)

4.2.1. Operation: Network Scanning to Gather Valve Actuator Data

The Network Master polls and controls up to 60 valves on the network by sequentially polling (scanning) each device in sequence from slave actuator address #1 to the last slave address (#n) on its network.
This polling gathers information from the actuators (including alarms, discrete information and actuator position) and places it in tables that can be transmitted to the PLC.

The scan operation cannot stop in the middle of a request for data from a particular actuator. Therefore, the 1746-C can only interface with the PLC once it has finished all required communications with the actuator it is currently getting data from. Network scanning resumes with the next actuator after interfacing with the PLC and processing any requests made by the PLC (writes to tables, writes to actuators, table reads, …).

For example, if the 1746-C is currently scanning actuator [5] and the PLC wants to write a new valve position setpoint to 22 actuators then the 1746-C will …

1. Finish getting data from actuator [5]
2. Get the new command table from the PLC.
3. Write the new valve position setpoint to all 22 actuators.
4. Restart network scanning where it left off … the next actuator [6] after the last one polled [5].

Standard data is gathered from the actuators by a single Modbus 03 command.
However, a 2nd Modbus 03 command is generated if any of the following are true …

- If requesting totalizer data
- If requesting a specific holding register
- If requesting a block of holding registers from this actuator
- If requesting to monitor the discrete outputs on this actuator
- If requesting TEC2000 input data.

This is one reason why additional data should only be requested when needed … it will slow the entire scan time of the system (by a minimum of 30-50 ms each time an actuator is polled for this data).
For instance, if all 5 of the above are requested for each actuator, an extra 30-50 ms is required for EACH … meaning that the scan time could be increased by almost a factor of 5! … dramatically slowing down the system!

A typical scan time (time to scan all the actuators on the network) to gather “standard” information on 60 actuators is less than 10 seconds … depending upon the 1746-C configuration.

When gathering data, the data is stored in the appropriate tables in the 1746-C. Values in the table for a particular actuator will not be overwritten if there is an error communicating (on both ports) with the actuator.

4.2.2. Operation: The Network Communication Diagnostic Scan

After a predetermined number of loops through the network scanning the actuators (the "scan period"), the 1746-C performs a communication diagnostic scan to determine if any "off-line" actuators can come back on line. If there are any that are ready, the 1746-C will bring them back on line without human intervention. This is the only time an off-line system can come back on-line.

The number of loops around the network is determined by the "scan period" value. The PLC adjusts this value to accommodate site requirements and current operating environments.

Unlike the other operations that access the actuators on the network, the diagnostic scan checks both ports when communicating with the actuators. Because of this, it also always takes longer to execute than standard data gathering scans. However, normal data is still gathered during a diagnostic scan.

The only major difference in data gathering between a standard scan and a diagnostic scan is

- In the diagnostic scan, the 1746-C determines actuator system type and actuator firmware version ID (tables [18 & 19]).
- Both ports are explicitly tested when getting the system type information.
- The diagnostic scan ensures that the hardware ports are set correctly … 9600 - 8N1.

Otherwise, all other data gathering activities are the same as in a normal network scan.

Communication error indications may only be cleared during this scan.

The "scan loop counter" is always zero (0) during a diagnostic scan … the diagnostic scan does not count as part of the scans in the scan period. This is the only time the loop counter is zero … during the diagnostic scan.

The diagnostic scan time increases as you increase the discrepancy between the

(number of valves actually on line and talking) VS (the number of valves configured to be on line)

Below shows a sample of diagnostic scan times empirically taken on a network when there was no network "writes to the actuators" (Table [0] was configured for 60 valves). As you can see, as you increase the number of "off line" units, you increase the diagnostic scan time.

# Nodes "Talking" … On-Line	Version 5.21 Diagnostic Scan Times (secs)
60 of 60 (#1 – 60)	15
54 of 60 (#7 – 60)	16
48 of 60 (#13 – 60)	17
42 of 60 (#19 – 60)	18
36 of 60 (#25 – 60)	19
30 of 60 (#31 – 60)	20
24 of 60 (#37 – 60)	21
12 of 60 (#49 – 60)	23
6 of 60 (#55 – 60)	23
3 of 60 (#58 – 60)	24
2 of 60 (#59 – 60)	25
1 of 60 (#60)	25

The values in this table "rounded values" and have a precision of (+/-) 1 second.

Because of this, these numbers are only appropriate for estimates and trend analysis.

Notes:

- Though not much, the diagnostic scan time can also vary with the number of valves in the valve scan time slice.
- The greater the number of valves configured to be on line, the longer the diagnostic scan time … more valves to check.
- During normal operations, an operator can use the HMI to force the system into a diagnostic scan by temporarily changing the "scan period" value in table [0] to a value lower than the current scan loop counter. This will cause the program to enter a diagnostic scan as soon as this scan loop has finished.
- Because the diagnostic scan detects faulty communication ports and flags them, there can be a perceived increased performance in the normal network scans because "off-line" actuators have already been flagged and will be skipped.

4.3. PLC Interface Time-Slice (1746-C ↔ PLC Communication)

The primary interface for the PLC with the 1746-C module is with an array of 25 tables (labeled 0 → 24) stored in the 1746-C RAM. Each table is 64-words long and each word is 16-bits. From the perspective of the PLC interfacing with the network, some of the tables are "read only" and some are "read/write" (depending upon a command word).

Relative to M0/M1 transfers, from the 1746-C perspective, there is no difference in handling "table reads" or "table writes" … they both require a M0/M1 file transfer command.

The Relay Ladder Logic (RLL) in the PLC CPU interfaces with the 1746-C module with the M0/M1 file transfer mechanism and transferring fixed blocks of 64 words (a table) with the table number (command) in Word # [0] of each transfer.

- To send a table to the 1746-C (either as a write table command or to indicate a particular table to read back), the PLC must send the desired table in the data portion of the M0 file.

- To read back a table, the PLC must read the M1 file sent from the 1746-C.

Note: *Since the 1746-C must be mounted in the local chassis (in the same chassis as the CPU), the RLL cannot use the BTR/BTW block transfer command set (BTR/BTW commands cannot be used over the back plane in the SLC systems). Therefore, the RLL must implement the transfer using the "copy file" command … copying the M0/M1 files back and forth.*

Note: *Refer to the following Allen Bradley Documents:*

- **Allen Bradley SLC-500 Instruction Set Reference,**
  Publication 1747-RM001D-EN-P (November-2003)

- **Allen-Bradley SLC-500 BASIC User's Manual (1746-BAS & 1746-BAS-T)**
  Publication No. 1746-UM004A-US-P 2000

The interfacing between the PLC and the 1746-C is performed in the 1746-C module in a section of RAM that is used as an exchange buffer area. To prevent potential overwriting problems when performing transfers, there are two separate buffers in this exchange area:

- M0 file … Basic Input Buffer (BIB): PLC fills to send 64 words to the 1746-C.
- M1 file … Basic Output Buffer (BOB): 1746-C fills to send 64 words to the PLC.

This is the only address accessed by the PLC when executing the M0/M1 transfer commands (meaning that all reads & writes by the PLC are to and from these buffers).

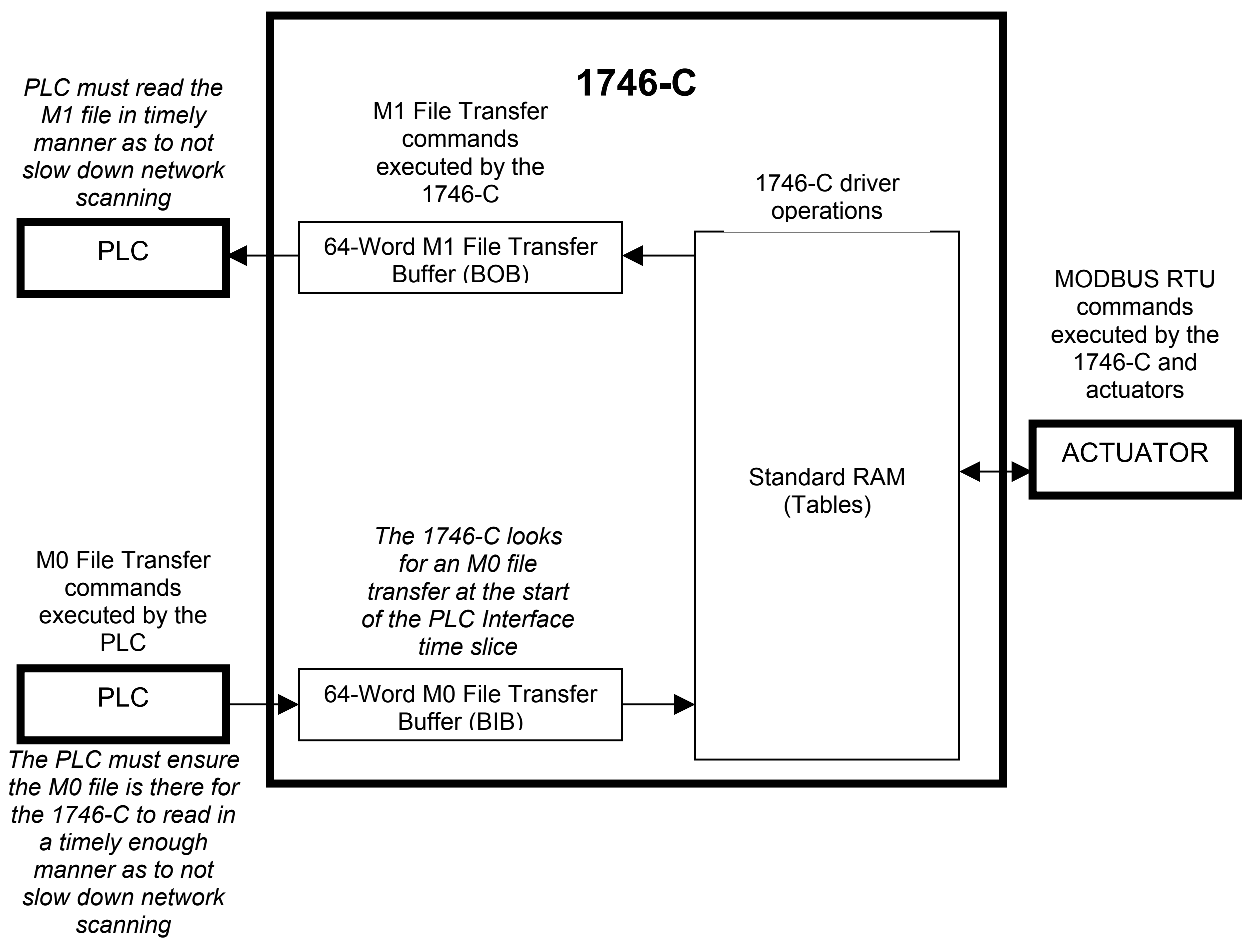


Figure 3-1 Concept Diagram of Command Operations

4.3.1. Operation: PLC (Write Data Tables To) or (Read Tables From) the 1746-C

Relative to M0/M1 transfers, from the 1746-C perspective, there is no difference in handling

- Writing a value to any table in the 1746-C memory.
- Writing a value to a table in the 1746-C memory that is also passed on to the specified actuator on the network.

They both require a M0 file transfer command. The only difference is in the 2nd word in the block being transferred (word [1]).

- If word [1] == 0, then this M0 transfer is a command to **read** a table from the 1746-C.
- If word [1] =/= 0, then this M0 transfer is a command to **write** data to a table in the 1746-C.

To perform a transfer of table data:

1. The PLC interface time-slice has now started.

2. The 1746-C looks to see if a new M0 file has arrived from the PLC. If not, this operation terminates and must be restarted on the next PLC interface time slice. Otherwise, continue.

3. Since a new M0 file is here, the 1746-C copies the table into its own memory.

4. If word [3] indicates an ESD command sequence (start or stop the ESD condition) then that is outputted to the entire network of actuators on both ports.

5. If word [1] == 0, then a read-only request command was made and the appropriate table is copied into the M1 exchange buffer.

   If word [1] =/= 0, then this was a write command and this table is copied into 1746-C memory. A final "resultant" table (what will be stored in the 1746-C and potentially sent out onto the network) is then copied into the exchange buffer to be fed back to the PLC.

6. The new M1 file is prepared to be transferred to the PLC such that if the received M0 file indicated
   - A read command – the data the PLC requested in a read table command.
   - A write command – the immediate feedback of the command structure the 1746-C is about to execute.

7. The 1746-C looks to see if the last M1 file was already read.

8. The new M1 file is transferred after the PLC indicates that the previous M1 file was read or the 1746-C system times waiting for the previous M1 to be read. Note that if the 1746-C times out waiting, the previous M1 file will be overwritten.

9. After the M1 file is sent, if the command was to write data to the actuators, the 1746-C will now loop through and write to all "appropriate" actuators on the network.

10. The PLC interface time-slice is now finished.

As per affecting network performance, **unless the command is to write data to the actuators**, the PLC does not have to limit the rate in which these commands are sent to the 1746-C because the 1746-C only sends back the table information it already has stored in memory. Conceptually, these do delay the overall scan time. However, the amount of delay is very minor from a global perspective. (*Obviously, the PLC must still ensure all appropriate M0/M1 TRANSFER handshaking is performed*)

As per writes that send data to the actuators, the rate should be limited. Please refer to the section that discusses writing data to the actuators on the network.

Note 1: *All tables can be read by the PLC. To ensure you only read from a read/write table, access it with the command word set to "0" (zero).*

Note 2: *An invalid table ID sent to the 1746-C will default to a read request for table [0] data.*

Note 3: *If the 1746-C times out waiting for the previous M1 file to be read by the PLC, the previous M1 file will be over-written.*

Note 4: *On write commands from the PLC, reading the immediate feedback is optional for the PLC. In other words, if the PLC has no need to inspect this table, then it can read it back but just throw it out. However, servicing the M1 transfer request is important to overall system throughput.*

For instance, the PLC program might be architected such that it treats an output pipe ("command") as just that … a one-way pipe that has no feedback. In this case, the PLC would use other mechanisms for feedback (ex: other tables) and does not require the redundant feedback on write commands.

Note 5: *When a request for an M1 transfer is made by the 1746-C, the 1746-C program waits (in the respective request function) until either the transfer completes or until a timeout is triggered (currently set at 2 seconds).*

This extended timeout time is to assist a PLC program that is having other (hopefully temporary) difficulties and might take a while to return to normal behavior.

Therefore, to not slow down 1746-C network scanning activities, the PLC-RLL needs to service these requests as soon as possible … even if it is to only throw away the table read back. Servicing the request for an M1 transfer made by the 1746-C is critical for overall timing efficiency.

Note 6: *One main point to remember on M0/M1 transfer timing is that once the 1746-C responds to an M0 transfer, the 1746-C will not return to process (setup for) another M0 file from the PLC until it:*

- *responds with an immediate feedback M1 file transfer request containing the table being addressed.*
- *finishes writing the entire table of values to the valves (if a write command).*
- *completes the set of scans in the next network scan time slice*

Note 7: *Once the 1746-C reads in a table from the M0 file, the 1746-C will always deliver the feedback table almost immediately … whether or not it was a read or write table command. Therefore, the best practice to maintain synchronization between the PLC and 1746-C is for the PLC to "conceptually" wait for a feedback response from the 1746-C before continuing … even if only to throw away the table.*
This will also increase the 1746-C overall speed because it waits for an acknowledgement of the feedback before continuing with command execution.

REMEMBER … Speed in servicing of the M1 transfer requests expedites system throughput.

4.3.2. __Operation: PLC Commands to Write Data to the Actuators__

When valve control commands (change setpoint, open, stop or close) are generated by the PLC, the Network Master sends the appropriate command(s) over the network to the addressed slave device (actuator).

However, if the command for a particular valve has not changed (value in the table is the same as before), then nothing is sent to the particular valve (it is skipped). Basically, when the 1746-C starts acting on a command to write data to the actuators, it compares the current value in the table with the new one. If the values are different, then the actuator is flagged as one to write information to.

> When writing discrete output control commands (directly controlling coils [3-5] or [100-1011]), the 1746-C does NOT compare the value to the last one. When commanded to write to the coils, it is performed ONE TIME and then turned off for that actuator (however, monitoring the status of those outputs is left on for that actuator).

When the 1746-C starts writing data to the actuators, is starts with the lowest address number and proceeds through to the highest address number. This write operation is completed after all “appropriate” actuators have been written to.

In general, there is one Modbus command/response set issued for EACH PLC command to write data to a particular actuator (ex: writing analog output data).

However, as per the specification, when a position control command is issued, the 1746-C is required to ensure that a “SSR” contactor configuration bit is set on any 320A system prior to receiving a control command. Therefore, there are up to 3 Modbus command/response sets issued for EACH PLC command to write position data in any particular 320A actuator:
If the actuator is configured to have an SSR in it (table [13]):

1. A Modbus command is sent to the actuator requesting status information to verify the SSR configuration coil is set.

2. If the coil is not set, a Modbus command is sent to set it.

3. The Modbus command is sent to indicate the new setpoint.

Note 1: *Discrete OPEN / CLOSE / STOP commands.*

> *The discrete OPEN / CLOSE / STOP commands (in table 3) are handled exactly the same as if a setpoint was sent.*
>
> - *OPEN (send a 100% open setpoint)*
> - *CLOSE (send a 0% open setpoint)*
> - *STOP (send a setpoint matching the last position read)*
>
> *(For more information on how this works, refer to the actuator’s operation manual)*

Note 2: *If writing to a valve and the write fails, then the internal flag to write the value REMAINS ACTIVE. This means that even though the communication process will prevent the program from trying to transmit when there is an error, the program will keep this value identified as one to send out until it is actually sent out or cleared by the PLC.*

This also means that the write to this particular actuator will be attempted the next time the PLC sends this table in with a "write table" command associated with it.
For example … If this was a value in table [5], the 1746-C will retry writing the value to the actuator the next time the PLC sends table [5] in as a write command.

The 1746-C cannot "cancel" this command because it is not in its scope of responsibility to directly change process control commands.
Therefore, it is the PLC's responsibility to write whatever "safe" value is appropriate for the particular actuator/application at any time in case the actuator "comes back online".

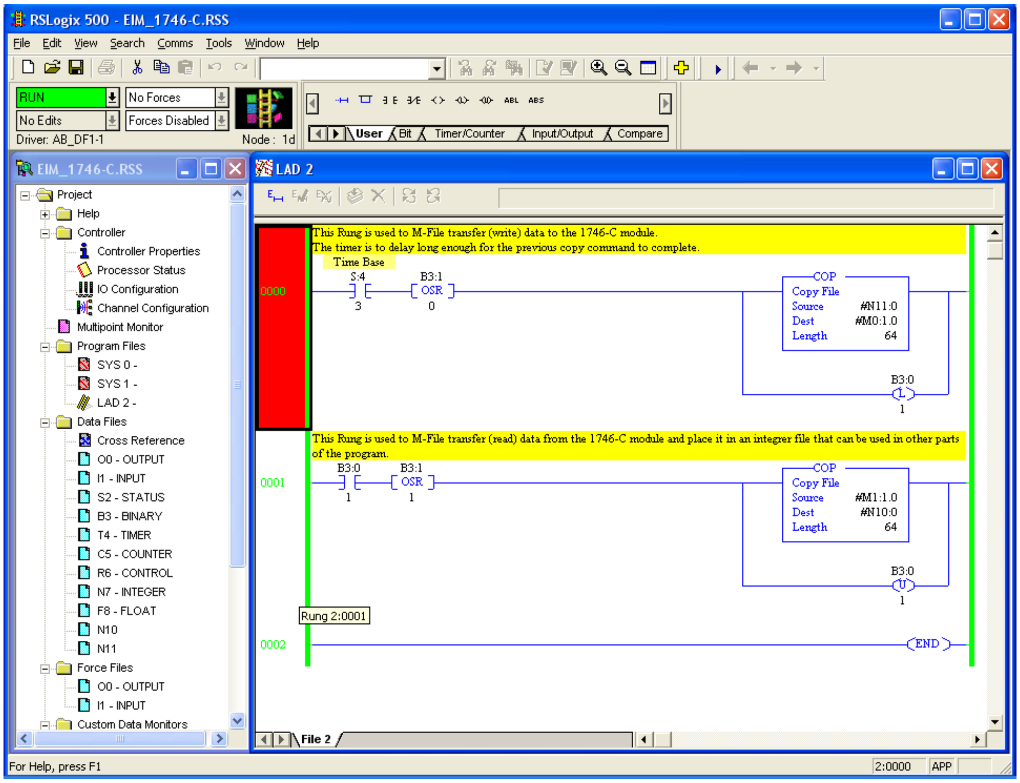
Note 3: *The PLC should limit the rate in which these commands are sent to the 1746-C … only write as fast as required to adequately control the valve. The faster the PLC requests changes to valve setpoint positions in the valves, the longer it takes the 1746-C to complete the write operation and continue scanning the network for data … decreasing the overall performance concerning updating status information for all actuators on the network.*

<u>For example</u>: Sending write commands once per second might still afford good control and still update all status information. However, sending write commands 4 times a second might write the information out faster but delay the response (status) information for the entire network to a point where it is too difficult to control.

Warning: *If the valve has been moved while the PLC is down or the valve was moved while in local mode, then as soon as the valve is communicating and accepting remote commands, it will move according to values in tables [3] and [5]. Please adjust the PLC RLL accordingly.*

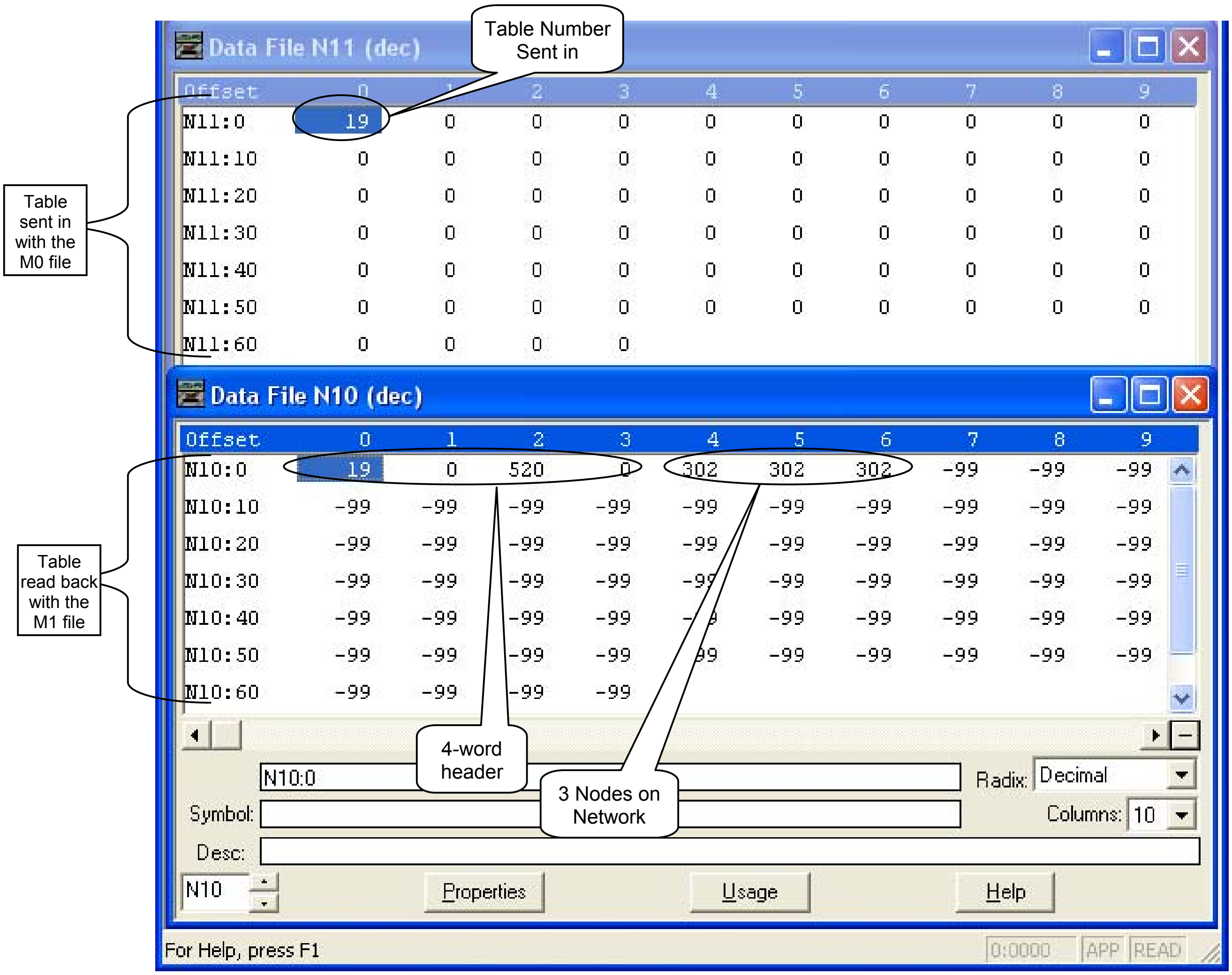
4.3.3. RLL Example … Manually Send New Table … Auto Read Response

Below is a simple example of an RLL that will manually send a table in a BTW and read back the resulting feedback table using a BTR.



On the next page is an example of the M0 & M1 files (tables) sent and received.

Control & Feedback via the N10 & N11 files … (table [19] requested … only 3 actuators on network)



5. System Tables

From the perspective of the PLC, all tables can be read but only some can be written to. If the PLC attempts to perform a write to a "read-only" table, the 1746-C ignores it and treats it as a read command/request.

In general, when writing data to be transmitted to the actuators over the network, the Network Master will differentiate which values change in tables being received from the PLC and will only send new command information to the actuators that require a change.

The tables have the following PLC ("user") access:

- Table [0]: read / write
- Table [1]: read only
- Table [2]: read only
- Table [3]: read / write
- Table [4]: read only
- Table [5]: read / write
- Table [6]: read only
- Table [7]: read only
- Table [8]: read / write
- Table [9]: read / write
- Table [10]: read / write
- Table [11]: read only
- Table [12]: read only
- Table [13]: read / write
- Table [14]: read / write
- Table [15]: read / write
- Table [16]: read / write
- Table [17]: read / write
- Table [18]: read only
- Table [19]: read only
- Table [20]: read / write
- Table [21]: read / write
- Table [22]: read / write
- Table [23]: read / write
- Table [24]: read / write

Table [0] is a "system focused" table. It is focused on information about the network system and network master. All the rest of the tables are focused on information about individual actuator "units".

NOTE: In tables [1 → 24 … except 15], the last 60 words hold information for particular actuators. Words [4 → 63] are in sequence with valve actuator network node addresses. A particular actuator is represented by its **POSITION** in the table as an offset starting from word [3] (the end of the header). For instance, Word [7] in all tables will always have valve #4 information.

There are two parts to a table:

- General Header Information: Words [0 → 3]
- Table Specific Information: Words [4 → 63]

In table [0], words [4 → 63] are divided into three parts:

- Words [4 → 17]: System Configuration Information
- Words [18 → 46]: -reserved- … currently not used.
- Words [47 → 63]: Run-time information inserted by the 1746-C.

5.1. Table Arrangement

Word offset into table	Table [0] ---- System Info Table ---- Read / Write	Table [1] ---- Comm. Error Status ---- Read Only	Table [2] ---- Interpreted Valve Status Bits Reg [06] ---- Read Only	Table [3] ---- Valve Control ---- Discrete Control ---- (O / C / S) ---- Read / Write	Table [4] ---- Valve Position Feedback ---- 0 – 100% in 0.1% increments ---- Read Only	Table [5] ---- Valve Control ---- Valve Setpoint ---- 0 - 4095 ---- Read / Write	Table [6] ---- Analog Input #1 ---- Read Only	Table [7] ---- Analog Input #2 ---- Read Only	Table [8] ---- Analog Output #1 ---- Read / Write	Table [9] ---- Totalizer #1 ---- Read / Write	Table [10] ---- Totalizer #2 ---- Read / Write	Table [11] ---- Valve Position 0 – 4095 reg [14] ---- Read Only	Table [12] ---- Status Bits for Discrete Digital Input s reg [05] ---- Read Only	Table [13] ---- SSR Config ---- Read / Write	Table [14] ---- Additional Register to Poll from All Valves on Network Read / Write	Table [15] ---- Block of Registers polled from a particular valve. Read / Write
0	Table # 0x00	Table # 0x01	Table # 0x02	Table # 0x03	Table # 0x04	Table # 0x05	Table # 0x06	Table # 0x07	Table # 0x08	Table # 0x09	Table # 0x0A	Table # 0x0B	Table # 0x0C	Table # 0x0D	Table # 0x0E	Table # 0x0F
1	R / W Command	0	0	R / W	0	R / W	0	0	R / W	R / W	R / W	0	0	R / W	R / W	R / W
2	**Reserved (F/W ID**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**
3	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**
4	**# Valves on Network**	Valve #1 Comm Status	Valve #1 Status	Valve #1 AOCS	Valve #1 POS 0 – 100.0%	Valve #1 STPT	Valve #1 AIN-1	Valve #1 AIN-2	Valve #1 AOUT	Valve #1 TTL - 1	Valve #1 TTL - 2	Valve #1 POS 0 – 4095	Valve #1 Status	Valve #1 SSR	Valve #1 Add Reg Value	**Valve #**
5	**Scan Period Size**	Valve #2 Comm Status	Valve #2 Status	Valve #2 AOCS	Valve #2 POS 0 – 100.0%	Valve #2 STPT	Valve #2 AIN-1	Valve #2 AIN-2	Valve #2 AOUT	Valve #2 TTL - 1	Valve #2 TTL - 2	Valve #2 POS 0 – 4095	Valve #2 Status	Valve #2 SSR	Valve #2 Add Reg Value	**Valve (Actuator) Type**
6	**Poll Totalizers**	Valve #3 Comm Status	Valve #3 Status	Valve #3 AOCS	Valve #3 POS 0 – 100.0%	Valve #3 STPT	Valve #3 AIN-1	Valve #3 AIN-2	Valve #3 AOUT	Valve #3 TTL – 1	Valve #3 TTL - 2	Valve #3 POS 0 – 4095	Valve #3 Status	Valve #3 SSR	Valve #3 Add Reg Value	**Starting Register Number**
7	**See table [0] Configuration & Run-Time Information**	Valve #4 Comm Status	Valve #4 Status	Valve #4 AOCS	Valve #4 POS 0 – 100.0%	Valve #7 STPT	Valve #4 AIN-1	Valve #4 AIN-2	Valve #4 AOUT	Valve #4 TTL - 1	Valve #4 TTL - 2	Valve #4 POS 0 – 4095	Valve #4 Status	Valve #4 SSR	Valve #4 Add Reg Value	**Number of Registers**
~		~	~	~	~	~	~	~	~	~	~	~	~	~	~	**See Table [15] Description**
63		Valve #60 Comm Status	Valve #60 Status	Valve #60 AOCS	Valve #60 POS 0 – 100.0%	Valve #60 STPT	Valve #60 AIN-1	Valve #60 AIN-2	Valve #60 AOUT	Valve #60 TTL - 1	Valve #60 TTL - 2	Valve #60 POS 0 – 4095	Valve #60 Status	Valve #60 SSR	Valve #60 Add Reg Value	

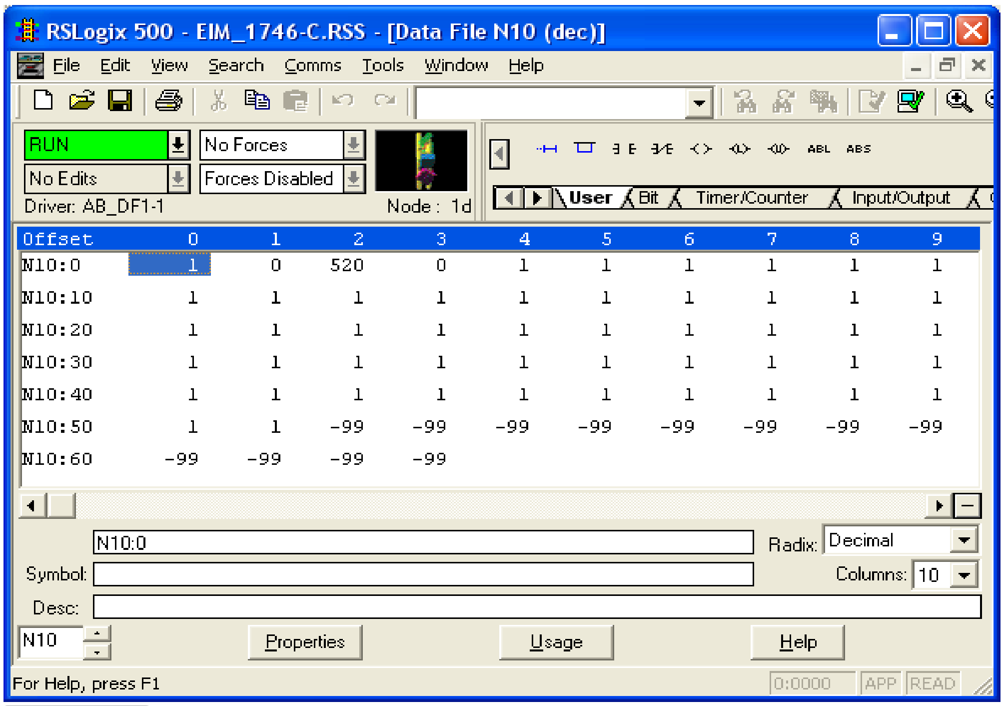
Word offset into table	Table [16] ---- Command to Monitor &/or Control Discrete Digital Outputs Read / Write	Table [17] ---- Status of Discrete Digital Outputs … Read / Write	Table [18] ---- System Type Read Only	Table [19] ---- Firmware Version ID Read Only	Table [20] ---- Modbus Exception Response Indications Read / Write	Table [21] ---- TEC2000 Inputs [1000 – 1015] Reg [1000] … Read / Write	Table [22] ---- TEC2000 Inputs [1016 – 1031] Reg [1001] … Read / Write	Table [23] ---- TEC2000 Inputs [1032 – 1047] Reg [1002] … Read / Write	Table [24] ---- TEC2000 Inputs [1048 – 1063] Reg [1003] … Read / Write
0	Table # 0x10	Table # 0x11	Table # 0x12	Table # 0x13	Table # 0x14	Table # 0x15	Table # 0x16	Table # 0x17	Table # 0x18
1	R / W	0	0	0	R / W	R / W	R / W	R / W	R / W
2	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**	**Reserved (F/W ID)**
3	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**	**ESD**
4	Valve #1 Discrete Output Cmd	Valve #1 Discrete Out Status	Valve #1 System Type	Valve #1 Firmware ID	Valve #1 Exception Code	Valve #1 Inputs [1000-1015]	Valve #1 Inputs [1016-1031]	Valve #1 Inputs [1032-1047]	Valve #1 Inputs [1048-1063]
5	Valve #2 Discrete Output Cmd	Valve #2 Discrete Out Status	Valve #2 System Type	Valve #2 Firmware ID	Valve #2 Exception Code	Valve #2 Inputs [1000-1015]	Valve #2 Inputs [1016-1031]	Valve #2 Inputs [1032-1047]	Valve #2 Inputs [1048-1063]
6	Valve #3 Discrete Output Cmd	Valve #3 Discrete Out Status	Valve #3 System Type	Valve #3 Firmware ID	Valve #3 Exception Code	Valve #3 Inputs [1000-1015]	Valve #3 Inputs [1016-1031]	Valve #3 Inputs [1032-1047]	Valve #3 Inputs [1048-1063]
7	Valve #4 Discrete Output Cmd	Valve #4 Discrete Out Status	Valve #4 System Type	Valve #4 Firmware ID	Valve #4 Exception Code	Valve #4 Inputs [1000-1015]	Valve #4 Inputs [1016-1031]	Valve #4 Inputs [1032-1047]	Valve #4 Inputs [1048-1063]
~	~	~	~	~	~	~	~	~	~
63	Valve #60 Discrete Output Cmd	Valve #60 Discrete Out Status	Valve #60 System Type	Valve #60 Firmware ID	Valve #60 Exception Code	Valve #60 Inputs [1000-1015]	Valve #60 Inputs [1016-1031]	Valve #60 Inputs [1032-1047]	Valve #60 Inputs [1048-1063]

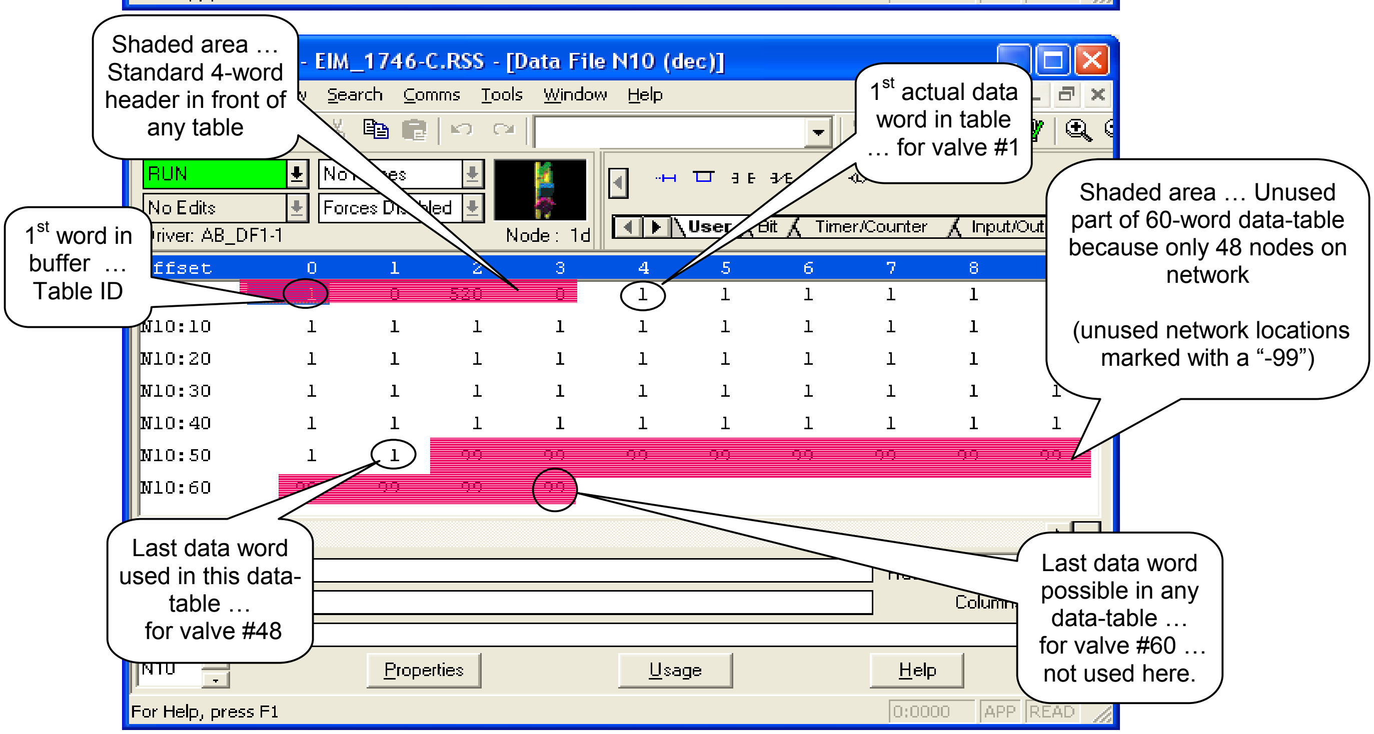
Figure 5-1 Table Arrangement

EXAMPLE OF GENERAL TABLE FORMAT …

Characteristics:

- 48 node network
- Table [1] feedback from the 1746-C (what is given to the PLC)





5.2. Standard Header Format (All Tables) … Words [0 → 3]

The header is the first four (4) words in the data block.
Please refer to the list of tables in Figure 3-2.

5.2.1. Word [0]: Table ID

As in the first word in the block data transfers, this first word is the Table ID. This can also be considered the "Command ID".

5.2.2. Word [1]: Read / Write Command Indicator

(If a Read/Write Table):

If word [1] == 0 this is a command from the PLC to the 1746-C to read the specified table from 1746-C memory and place it into the exchange buffer so the PLC can read it out … words [0 → 63].

If this is table [0], then this read back will include all current configuration information (at the front of the table) and current system run-time information (at the back of the table).

If word [1] =/= 0 (any non-zero value) … this is a command from the PLC to the 1746-C to remove the supplied table of data from the exchange buffer and place it into the 1746-C standard memory (tables).

- If this is table [0], then the configuration information is written.
- Otherwise, the table information is written to the 1746-C.
- If this is the type of table that actually writes data to the actuators, then that data is outputted to the actuators.

(If a Read Only Table):

This word is ignored and the table is only read.

5.2.3. Word [2]: Reserved / Firmware ID

When received by the 1746-C from the PLC, this word is ignored (reserved for future use).

When transmitted from the 1746-C to the PLC, this word contains the firmware ID. This is a decimal number representing the 1746-C firmware ID. For instance, if the value in table [0] … word [2] is 400, the firmware version number is 4.00

5.2.4. Word [3]: Emergency Shut Down (ESD) Command Word & Indication

The ESD command is a command broadcasted to all actuators connected to the network that an emergency is in effect and the actuator should start performing the ESD actions its pre-configured for. The 1746-C broadcasts an ESD on both channels with a 200-millisecond delay between them.

If the PLC wishes to engage the ESD for the network, it writes a 255 (0x00FF) to this word location (Word [3]) using a write to any table. The Network Master then broadcasts the appropriate ESD command to all valves on the network (on both channels with about a 200ms delay between each).

When the PLC wants to end the ESD, it writes a 170 (0x00AA) into this word into any table. The Network Master will then broadcast the appropriate "Stand-Down from ESD" command to all valves on the network on both channels with about a 200ms delay between each.

Later, after the PLC is satisfied that the ESD is over and normal operations have resumed (stand down from ESD has completed), it is recommended that the PLC clear this stand-down value (and all aspects of ESD) by resetting it to zero (again, in any table).

Summary: The appropriate values for the ESD command word from the PLC are:

- 255 (0x00FF) = Engage ESD
- 170 (0x00AA) = Stand down from ESD
- 0 = Emergency shut-down (ESD) control is not active

Once any ESD command is written to any table, the value is propagated throughout all tables. Therefore, the RLL does not have to remember which one it wrote to.

Warning: *Since the ESD is actually a function of the actuator control cards; if an actuator is reset then that actuator will probably not be in ESD. This also means that if the 1746-C loses power or is reset, it has no prior knowledge of any valves in ESD. It can turn off ESD once initialization of the software has occurred. It is the responsibility of the RLL to repeat the ESD broadcast under the possibility of these circumstances.*

Warning: *If any or some of the actuators are configured to move on ESD, consider the electrical power required in performing a network ESD. A large electrical power drop can possibly create a reset in the actuator by affecting the control voltage.*

Note 1: *Remember, an ESD command from the PLC will inform the actuators to do what ever they were configured to do during an ESD condition. Therefore, this could easily override any other control commands already sent to them or about to be sent (ex: commands placed in any write Tables [3, 5, 8,] might be rejected by the actuator because of the ESD condition).*

Note 2: *If monitoring the network, the values of 0xAA & 0xFF are not the values transmitted to the actuators. The program actually uses a Modbus command 05 to set/clear a bit (coil) in the register map in the actuators. From then on, the actuators do what ever they are configured to do when ESD is either started or ended.*

5.3. Table [0] … System Information Table

5.3.1. Table [0] Overview

This table is a read/write table in that is used for "system level" configuration and run-time status/diagnostic information. Basically, it is not "actuator" focused but more focused on the entire network of actuators.

When told to write this table, the information is not sent out onto the network. Also, only the configuration words are written into the table. The run-time status information is supplied by the 1746-C.

Table [0] has 4 parts to it:

1. Words [0 → 3]: Standard header information … common in all tables.
2. Words [4 → 17]: System Configuration/Control words (set by the PLC)
3. Words [18 → 46]: Not currently used … Reserved for future use … zeroed out.
4. Words [47 → 63]: Run-time feedback of real-time values (set by the 1746-C).

Words [4 → 17] (control/configuration parameters set by the PLC)

1. Word [4]: Set the Number of Valves on the Network.
2. Word [5]: Set the Scan Period value (# scans before a communication diagnostic scan)
3. Word [6]: Set the T/F control flag whether or not to poll for totalizers on network scans.
4. Word [7]: Set the additional number of valves to poll in the network-scan time slice.

5. Word [8]: Set the Additional Message response delay to expect from actuators (10ms increments)

(FUNCTION: Poll specified additional register from all actuators on network)

6. Word [9]: Set the specified additional register to poll from all actuators on the network.
7. Word [10]: Set the T/F control flag whether or not to start polling all actuators for the additional register.

(FUNCTION: Poll block of registers from one specific actuator on the network)

8. Word [11]: Set the specific actuator to get the block of holding registers from
9. Word [12]: Set the starting register number in the block of holding registers to be requested
10. Word [13]: Set the total number of holding registers being requested
11. Word [14]: Set the T/F control flag whether or not to start polling the specific actuator for the block of registers.

(FUNCTION: Reset Scan Period Counter/Accumulator)

12. Word [15]: Set the T/F control flag whether or not to manually reset the Scan Period Counter.
13. Word [16]: Value that will always reset the scan period counter.

(FUNCTION: Poll TEC2000 Input Registers 1000 – 1003)

14. Word [17]: Set the T/F control flag whether or not to poll the input registers if the system is determined to be a TEC2000 system.

Remember … Currently, Words [18 → 46] in table [0] are unused and should be considered “reserved” for future use.

Words [47→ 63] (Run-time feedback of real-time values set by the 1746-C)

1. Word [47]: Rollover Counter for Scan Period Counter (Accumulator).
2. Word [48]: Scan Period Counter (Accumulator).

----

3. Word [49]: The port number for the currently “preferred port”.
4. Word [50]: Total number of valves polled during last network-scan time slice (before servicing the PLC communications).

----

5. Word [51]: This scan period: Last Network Scan Time (0.1 sec increments).
6. Word [52]: This scan period: Total elapsed time (0.1 sec increments).
7. Word [53]: This scan period: Average scan time (0.1 sec increments).
8. Word [54]: This scan period: Shortest scan time (0.1 sec increments).
9. Word [55]: This scan period: Longest scan time (0.1 sec increments).

----

10. Word [56]: Last scan period: Average scan time last period (0.1 sec increments).
11. Word [57]: Last scan period: Total scan time last period (0.1 sec increments).
12. Word [58]: Last scan period: Shortest scan time last period (0.1 sec increments).
13. Word [59]: Last scan period: Longest scan time last period (0.1 sec increments).

----

14. Word [60]: Duration of last diagnostic scan. (0.1 sec increments).
15. Word [61]: Duration of previous diagnostic scan (0.1 sec increments).

----

16. Word [62]: Current period scan loop counter.
17. Word [63]: Current number of actuators “online” and communicating.

Note: All real-time timing information reported (reported back by the 1746-C module) has an accuracy of (+/-) 50ms.

5.3.2. <u>Table [0] Configuration Words … Words [4→ 17]</u>

The following words in table [0] are a configuration and control parameters set by the PLC for the 1746-C to run as expected.

Word [4]: Number of Actuators on the network.

This is value indicates of the number of actuators on the network. If this number ever changes, all tables [1-20 … except for 15] are reset to ensure valid values get filled for each valve on the network.

The minimum number of actuators is one (1).
The maximum is 60 (which is also the default).

Note to Remember with PLC Interfacing:

The 1746-C only fills in tables with data from the actuators during the "<u>network scan time slice</u>".
<u>If the PLC changes the Number of Actuators on the Network</u> *at some point in time after initial power up, the 1746-C will only be filling in the new data "X" actuators at a time (where "X" is the number of valves scanned per network scan time slide … see word [7] for more information on this). This means that it can take several network scan time slices for the 1746-C to fill in new data for all actuators on the network.*

The 1746-C will not attempt to fill in all data before interfacing with the PLC again because of the time it takes to poll the entire network … this could lead to "starving the PLC" with a lack of communication.

The PLC RLL needs to be constructed to take this into consideration.

Word [5]: Number of Scans in Scan Period … Scan Period "Size".

The scan period is the number of network scans before the network is retested for network failures (before the next diagnostic scan).

The default is 50 scans. If there is a communication error, such as a broken communication cable, then the system will look for a change (such as a repair to the cable) every 50 scans. The diagnostic scan resets all communication error indicators so that both ports can be checked

The minimum scan period number is five (5). The maximum number is 200.

Note: If your system requires highest performance and speed, then you may want to raise this value (200 is max). If the system does not require performance but requires high reliability and early communication fault detection, you might want to lower this number (5 is minimum).

Note: For more information on using the scan period value and the diagnostic scan; refer to the Application Note at the end of this document.

Word [6]: FUNCTION: Poll for Accumulator (Totalizer) Data

This is a flag that indicates whether or not the 1746-C should request accumulator (totalizer) data from the actuators when scanning. The totalizer values are stored in tables [9 & 10].

This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

If word [6] == 0 do not poll for totalizer data.
If word [6] =/= 0 (any non-zero value) … poll for totalizer data.

By default, word [6] == 0 … do not poll for totalizer data.

Word [7]: Additional Valves to Poll in the scan time-slice.

The process scheduler has 2 primary processes to monitor:

1. Scanning the network (polling valves)
2. PLC communication & service.

Each process is given a particular time slice to operate in/with.

By program architecture, it is guaranteed that after servicing a PLC request, one valve (the next on the network after the previous one polled) will be polled for data. The number in word [7] allows the operator to allocate additional valves to be polled before servicing the PLC again. The larger this number is, the shorter the network scan time but the fewer interactions with the PLC. The lower the number, the more service given to the PLC (for control and feedback) but the slower the network scan time (because we are servicing the PLC commands to read/write tables).

This number should be adjusted for the most optimal fit for the application.
The default value is 2.
The minimum value is zero (0).
The maximum value is the lesser of

- 9
- 1 less than the number of valves on the network.

Word [8]: Additional Message Response Delay

Word [8] offers the operator the chance to enter a desired ADDITIONAL expected delay when receiving messages from the network. The value is in 10 ms increments and can have a range of 0-30 (0 – 300 ms).

The default value is 0.

Words [9 & 10]: FUNCTION: Poll Additional Holding Register From Entire Network

This operation allows the operator the chance to poll the entire network for any particular holding register. The register values polled are stored in table [14]. This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

This data is retrieved with a separate Modbus read command each time an actuator is polled (scanned).

FORMAT:

Word [9] … The holding register desired.
Word [10] … This is a flag that indicates whether or not to start requesting that register from the actuators when scanning.

- If Word [10] == 0 do not poll for the register.
- If Word [10] =/= 0 (any non-zero value) … poll for the register specified in word [9].

By default, Words [9 & 10] == 0 … do not poll.

Words [11 → 14]: FUNCTION: Poll Additional Block of Registers From Single (Specified) Actuator

This operation allows the operator the chance to poll a single actuator for a block of registers. The register values returned are stored in table [15] such that …

- Table [15] – words [4-7] … header information about information returned.
- Table [15] – words [8-13] … not used … reserved.
- Table [15] – words [14 - 63] … register information returned.

This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

FORMAT:

Word [11] … The actuator address to get the block of registers from.
Word [12] … The register number for the 1st holding register in the block to retrieve.
Word [13] … The total number of registers to retrieve.
Word [14] … This is a flag that indicates whether or not to start requesting that block of registers from that actuator when polling it.

- If Word [14] == 0 do not poll for the block of registers.
- If Word [14] =/= 0 (any non-zero value) … poll for the block of registers.

By default, Words [11 → 14] == 0 … do not poll.

Words [15 → 16]: FUNCTION: Reset Scan Period Counter (Accumulator)

This operation allows the operator the chance to reset the scan period counter/accumulator

- When desired … word [15] =/= 0.
- With the value desired … word [16] value.

The scan period counter is located in table [0] – words [47 & 48] such that

- Word [47] … Rollover counter for the Scan Period Counter.
- Word [48] … Scan Period Counter

Both have a high end max value of 32,000.

Word [48] is incremented at the start of scan #1(after the previous diagnostic scan).
Whenever word [48] rolls over,

- Word [48] is automatically filled with the value in word [16].
- Word [47] is incremented

If the counter is reset in the middle of a scan period (which is most likely), it must wait until after that scan period is finished (and the accompanying diagnostic scan) before it starts counting again.

Note: that to determine the time for the rollover counter to rollover one time …

Rollover increment time = (time for the scan period) + (diagnostic scan time)

FORMAT:

Word [15] … This is a flag that indicates whether or not to reset the Scan Period Counter.
Word [16] … The value used to reset the Scan Period Counter.

- If Word [15] == 0 do not reset the counter.
- If Word [15] =/= 0 (any non-zero value) …
    - Word [47] = 0
    - Word [48] ← ← ← Word [16]

By default, Words [15 & 16] == 0 … do not reset the counter.

Word [17]: FUNCTION: Poll for Input Registers [1000-1003] on TEC2000 Systems

This is a flag that indicates whether or not the 1746-C should request the input registers on any TEC2000 system detected when scanning. The input register values are stored in tables [21 - 24] such that …

- Table [21] contains register [1000] status bits.
- Table [22] contains register [1001] status bits.
- Table [23] contains register [1002] status bits.
- Table [24] contains register [1003] status bits.

This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

If word [17] == 0 do not poll for input registers.
If word [17] =/= 0 (any non-zero value) … poll for input registers.

By default, word [17] == 0 … do not poll.

5.3.3. Table [0] Run-Time Information Words … Words [47→ 63]

The following words in table [0] are a run-time feedback of real-time values set by the 1746-C.

1. Word [47]: Rollover Counter for Scan Period Counter (Accumulator).
2. Word [48]: Scan Period Counter (Accumulator).

----

3. Word [49]: The port number for the currently “preferred port”.
4. Word [50]: Total number of valves polled during last network-scan time slice (before servicing the PLC communications).

----

5. Word [51]: This scan period: Last Network Scan Time (0.1 sec increments).
6. Word [52]: This scan period: Total elapsed time (0.1 sec increments).
7. Word [53]: This scan period: Average scan time (0.1 sec increments).
8. Word [54]: This scan period: Shortest scan time (0.1 sec increments).
9. Word [55]: This scan period: Longest scan time (0.1 sec increments).

----

10. Word [56]: Last scan period: Average scan time last period (0.1 sec increments).
11. Word [57]: Last scan period: Total scan time last period (0.1 sec increments).
12. Word [58]: Last scan period: Shortest scan time last period (0.1 sec increments).
13. Word [59]: Last scan period: Longest scan time last period (0.1 sec increments).

----

14. Word [60]: Duration of last diagnostic scan. (0.1 sec increments).
15. Word [61]: Duration of previous diagnostic scan (0.1 sec increments).

----

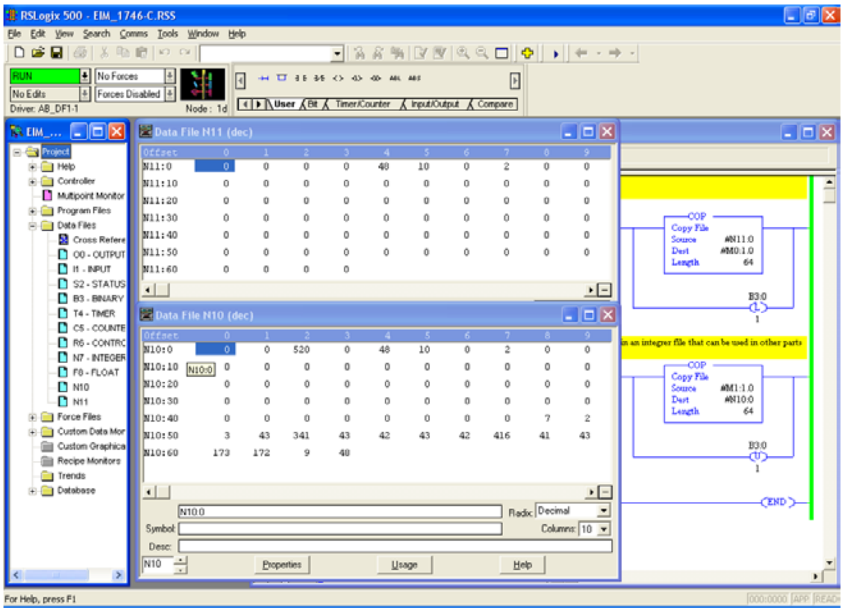
16. Word [62]: Current period scan loop counter.
17. Word [63]: Current number of actuators “online” and communicating.

This information is updated at the end of each scan. If the information is scan-period based, then the information is updated only at the end of the last scan for the particular scan period.

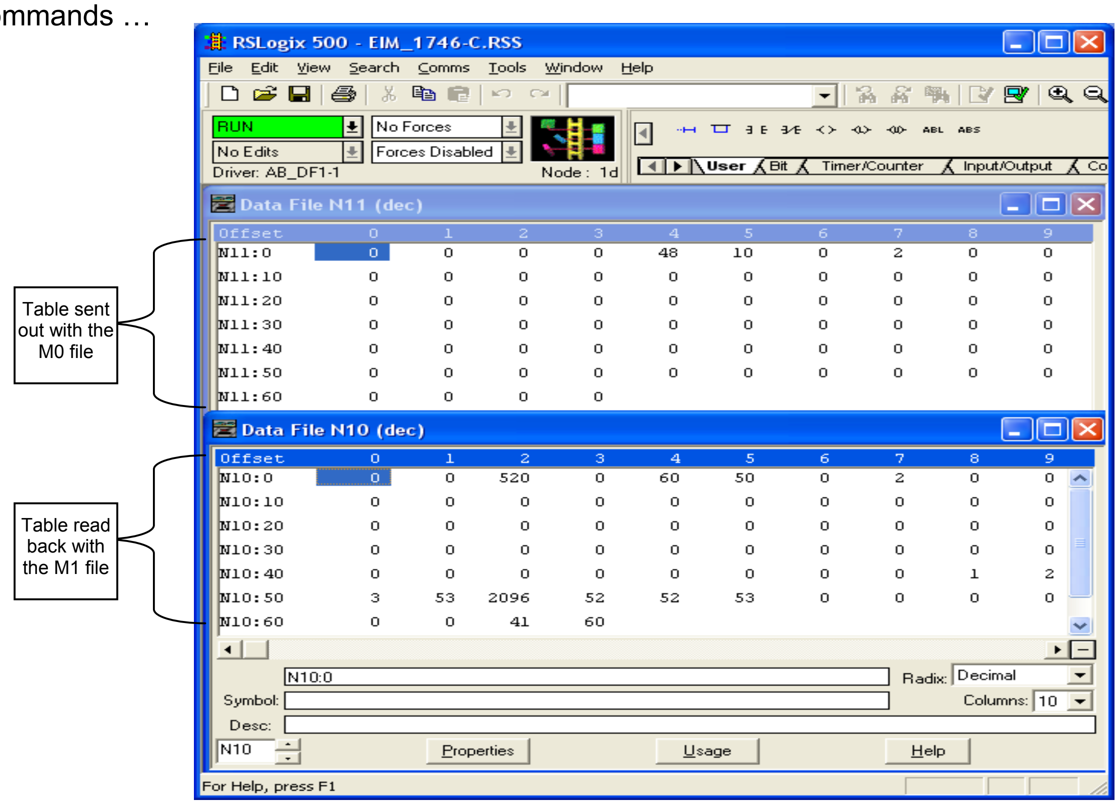
Note: All real-time timing information reported (reported back by the 1746-C module) has an accuracy of (+/-) 50ms.

5.3.4. Table [0] … RSLOGIX-500 Examples

5.3.4.1. Example RSLOGIX-500 Screen:



In this example, memory locations N10 & N11 show the entire data exchange area for M0/M1 transfer commands …



5.3.4.2. Table [0] … The Configuration Values (RSLOGIX-500).

In this example, the memory areas:

- N10:0 → N10:63: Indicates the 64-word table the PLC sends to the 1746-C.
- N11:0 → N11:63: Indicates the 64-word table the 1746-C sends back to the PLC.

Interesting & Important Note in this example:

In this example, the configuration the PLC would like to have (N10:0 → N10:16) does NOT agree with the configuration information the 1746-C actually has and is being read back (N11:0 → N11:16). Also notice the configuration values being reflected back by the 1746-C … they are all factory default values.

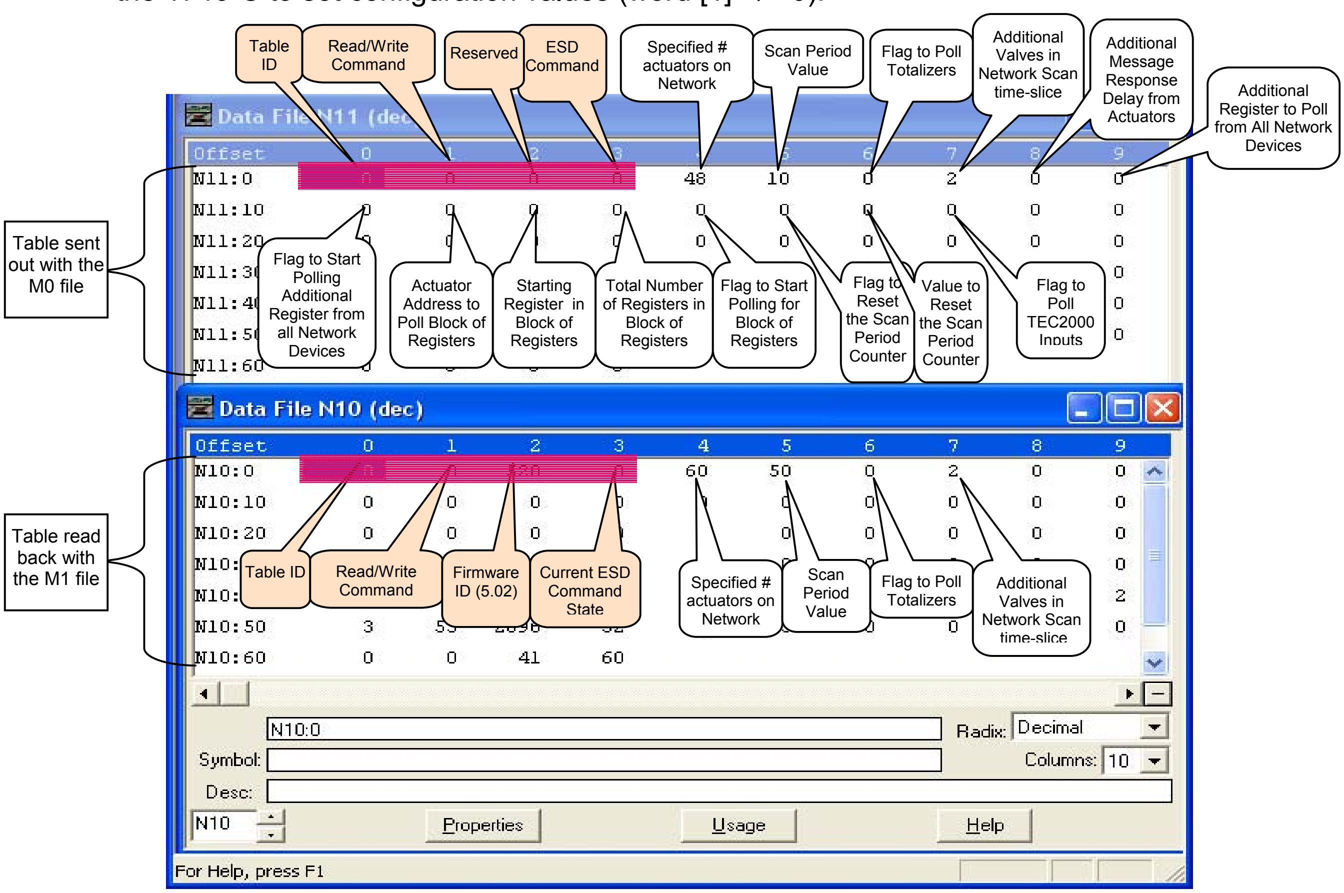
This screen reflects a state in which the PLC has not sent a “write table [0]” command to the 1746-C to set configuration values. Therefore, this 1746-C is currently running and “thinking” that

- There are 60 actuators on the network.
- The scan period should be 50 scans long

However, the PLC actually “wants”

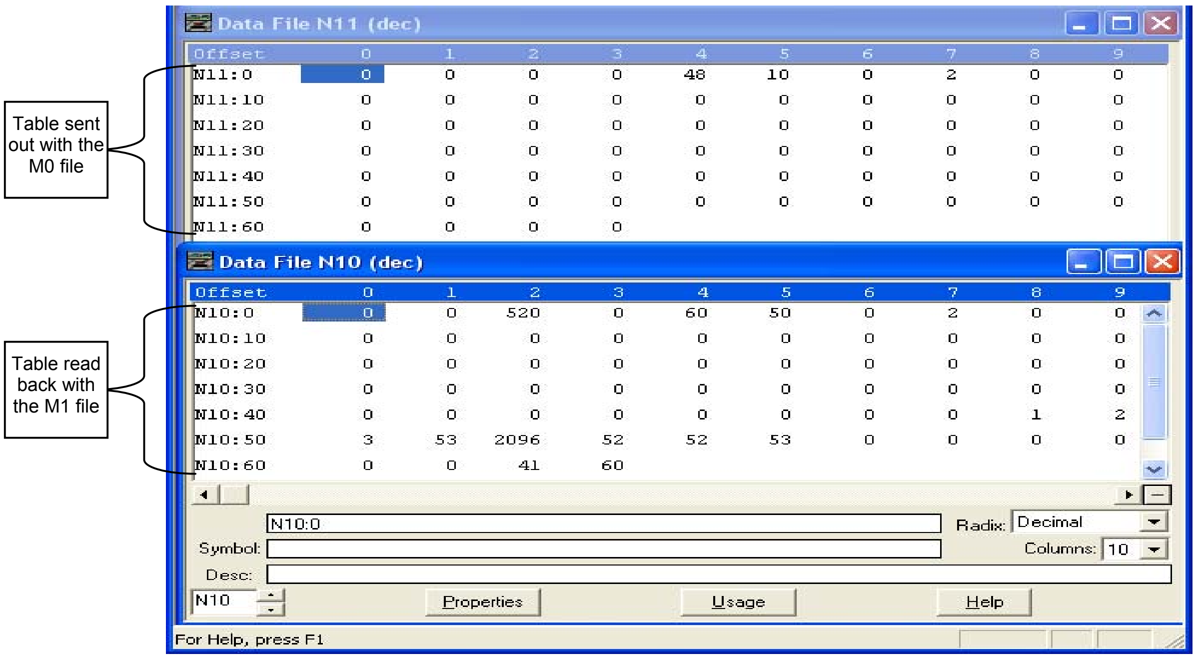
- 48 actuators on the network.
- A scan period 10 scans long

To remedy this disparity of operation … the PLC needs to send the “write table [0]” command to the 1746-C to set configuration values (word [1] =/= 0).



5.3.4.3. Table [0] … The Configuration Values – Writing Table [0] (RSLOGIX-500)

State prior to writing table [0] configuration values …



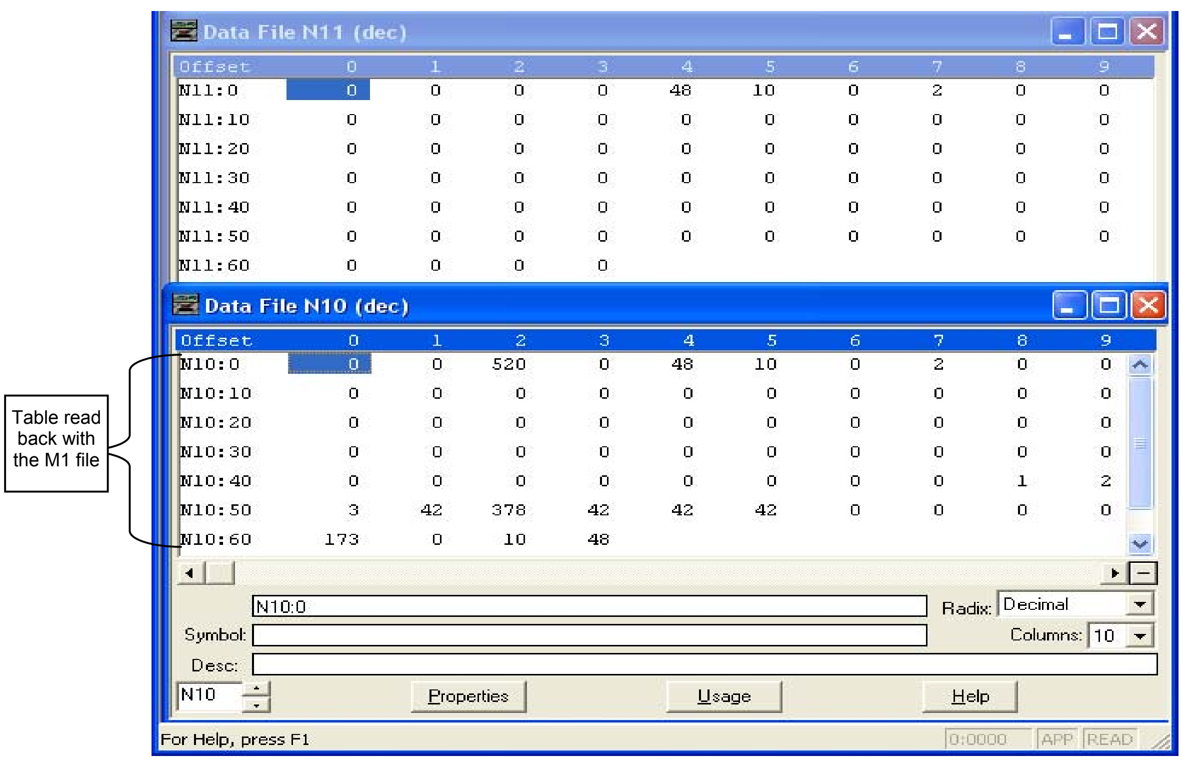
Data File N11 (dec)

Offset	0	1	2	3	4	5	6	7	8	9
N11:0	0	0	0	0	48	10	0	2	0	0
N11:10	0	0	0	0	0	0	0	0	0	0
N11:20	0	0	0	0	0	0	0	0	0	0
N11:30	0	0	0	0	0	0	0	0	0	0
N11:40	0	0	0	0	0	0	0	0	0	0
N11:50	0	0	0	0	0	0	0	0	0	0
N11:60	0	0	0	0						

Data File N10 (dec)

Offset	0	1	2	3	4	5	6	7	8	9
N10:0	0	0	520	0	60	50	0	2	0	0
N10:10	0	0	0	0	0	0	0	0	0	0
N10:20	0	0	0	0	0	0	0	0	0	0
N10:30	0	0	0	0	0	0	0	0	0	0
N10:40	0	0	0	0	0	0	0	0	1	2
N10:50	3	53	2096	52	52	53	0	0	0	0
N10:60	0	0	41	60						

State after writing table [0] configuration values …



Data File N11 (dec)

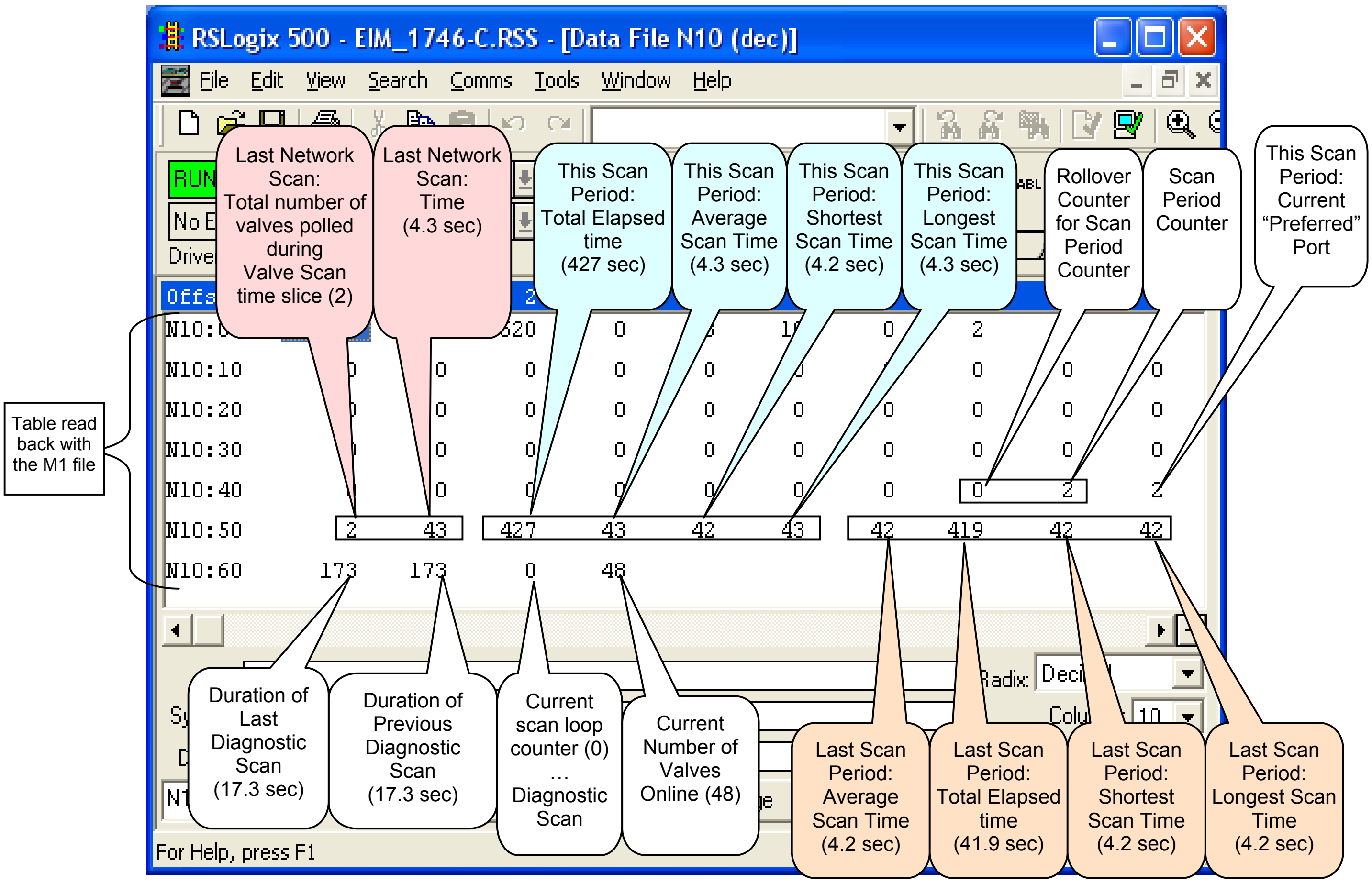
Offset	0	1	2	3	4	5	6	7	8	9
N11:0	0	0	0	0	48	10	0	2	0	0
N11:10	0	0	0	0	0	0	0	0	0	0
N11:20	0	0	0	0	0	0	0	0	0	0
N11:30	0	0	0	0	0	0	0	0	0	0
N11:40	0	0	0	0	0	0	0	0	0	0
N11:50	0	0	0	0	0	0	0	0	0	0
N11:60	0	0	0	0						

Data File N10 (dec)

Offset	0	1	2	3	4	5	6	7	8	9
N10:0	0	0	520	0	48	10	0	2	0	0
N10:10	0	0	0	0	0	0	0	0	0	0
N10:20	0	0	0	0	0	0	0	0	0	0
N10:30	0	0	0	0	0	0	0	0	0	0
N10:40	0	0	0	0	0	0	0	0	1	2
N10:50	3	42	378	42	42	42	0	0	0	0
N10:60	173	0	10	48						

5.3.4.4. Table [0] … The Run-Time Feedback Values From 1746-C (RSLOGIX-500)

This screen demonstrates examples of the run-time information that the 1746-C reports back to the PLC.



5.4. Tables [1 → 20] … Actuator Information Tables

5.4.1. Table [1]: Communication Error Status

Words [4 → 63] PLC access: Read Only.

The "Communication Error Status Table" is used by the network master to track communication faults with each actuator. This data is totally derived by the 1746-C. These are considered "Low-Level" communication errors.

The value of the word gives the following status:

- 0 = good communication through both ports
- 1 = error communicating through Port 1 … but success on Port 2
- 2 = error communicating through Port 2 … but success on Port 1
- 3 = No communication to the valve actuator at all. In this case, bit-14 in Table [2] will also be set.

This table can be used by the PLC system to diagnose valve actuators that are not communicating. If a network requires maintenance, then information in this table can help trace the problem. During a network fault, this information can help point to the section that's in fault.

During communication with the actuators, a time-out value is implemented to flag network errors. The network master will try to communicate with the valve from the currently "preferred port" three (3) times. If the valve does not respond within those three times, the valve is flagged to communicate on the current "backup port" and the 1st port is identified as having a problem with that valve (in table [1]). The backup port immediately tries to communicate with the valve three more times. If the valve still fails to respond, the particular actuator is flagged as having a general communication error by having

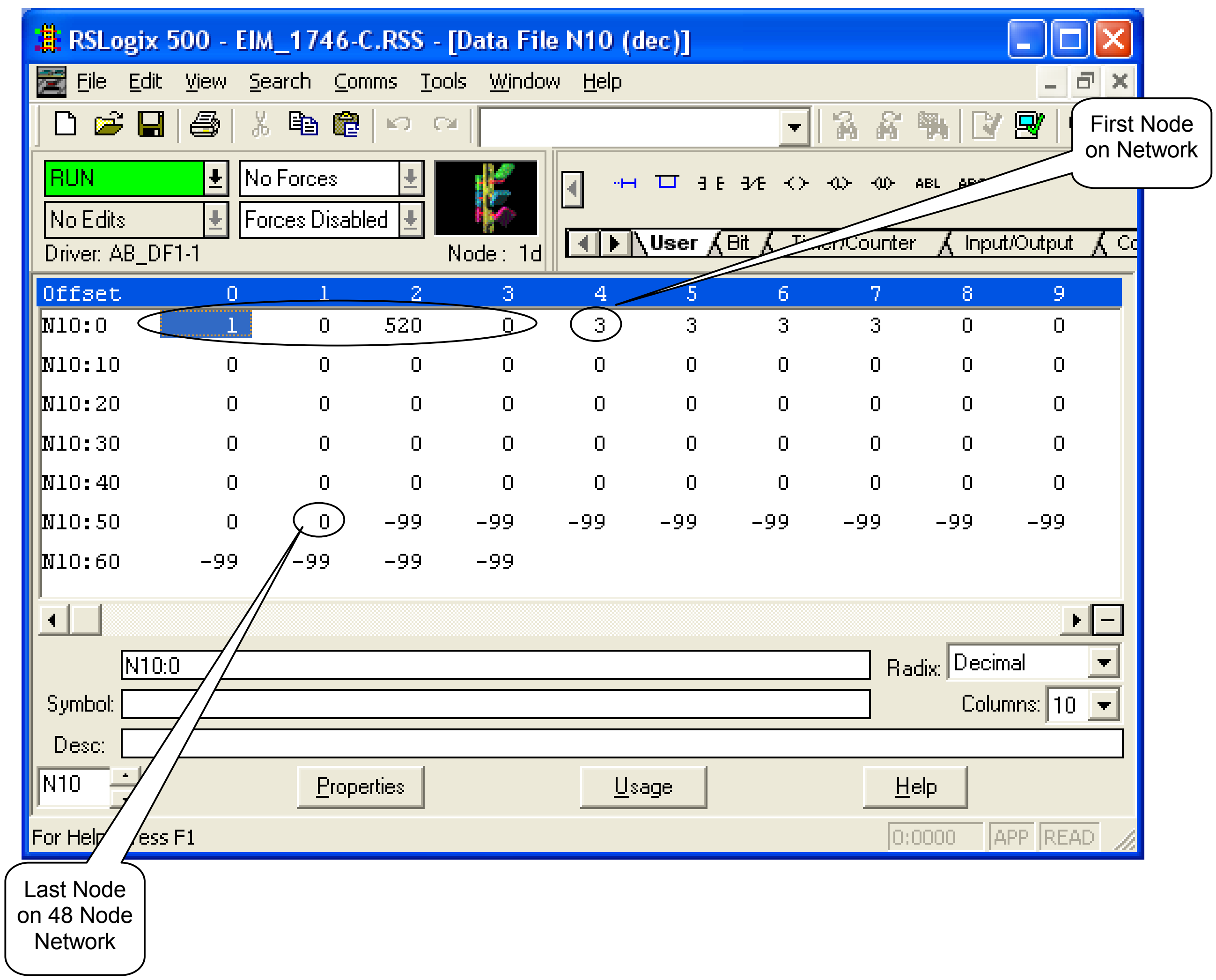
- Its table [1] entry set = "3".
- Its bit-14 in table [2] set = "1".

The data in this table is always updated … any time the 1746-C tries to communicate with the network. However, this table is only reset (errors cleared) during a diagnostic scan.

Example Table [1] Feedback (48-node network) …

Characteristics:

- Total COM faults on units [1-4] … they have a value of 3 … an error communicating on both ports.
- No COM faults on any other units (5 – 48 have a value of 0 … no problems communicating).
- Note that this would happen if units [1-4] were turned off.



5.4.2. Table [2]: Actuator Operational Status

Words [4 → 63]   PLC access: Read Only.

This data is primarily derived from holding register [6] in the actuator. It is retrieved with the "standard data request" - Modbus read command when the particular actuator is being polled (scanned). However, bit-14 is adjusted by the 1746-C.

This status word can be "bit parsed" to give:

- Bit-0: LSO is tripped (valve is at the end of its OPEN travel limits)
- Bit-1: LSC is tripped (valve is at the end of its CLOSE travel limits)
- Bit-2: Valve is moving in the OPENING direction
- Bit-3: Valve is moving in the CLOSING direction

- Bit-4: Selector Switch is in the Local position
- Bit-5: Selector Switch is in the Remote position
- Bit-6: Open Torque ALARM is active.
- Bit-7: Close Torque ALARM is active.

- Bit-8: Valve Stall Alarm is active (valve not moving on command)
- Bit-9: Power Monitor Alarm is active (loss of control voltage alarm)
- Bit-10: Motor Overload Alarm is active
- Bit-11: Phase Monitor Alarm is active (only if Phase Sentry is Installed)

- Bit-12: Local (Hardwired) Alarm is active.
- Bit-13: Fail Alarm Self-diagnostics Alarm is active.
- Bit-14: Comm Alarm … Valve not communicating with the network
  (Same as the value "3" in Table [1])

- Bit-15: Unit Alarm … Set when any actuator alarm is set.

Note: Bit-0 is the LSB in the word. Bit-15 is the MSB in the word.

Some of these bit indications are repeats of those in table [12]. However, not all.

Example Table [2] Feedback (48-node network) …

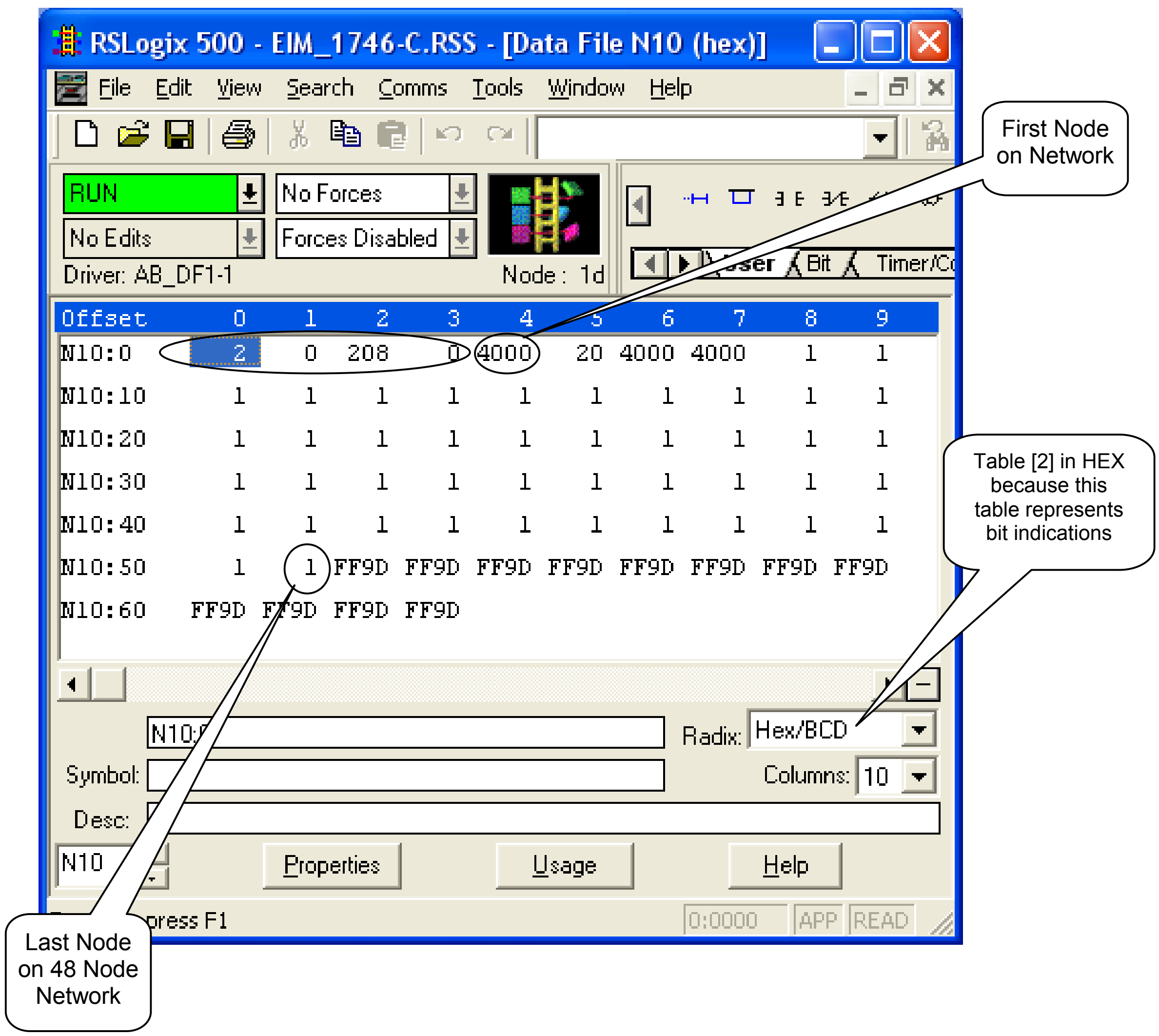
Characteristics:

- The format is displayed in HEX because HEX is easier to interpret "bit indicator" data.

- Units [1, 3, 4] …
  - Communication Alarm exists … bit [14] = 1

- Unit [2] …
  - Selector switch in the remote position … bit [5] = 1

- The rest of the network is all the same …
  - LSO is tripped … bit [0] = 1



5.4.3. Table [3]: Discrete Control Mode (Open/Close/Stop)

Words [4 → 63] PLC access: **Read / Write**

This table commands the actuator to move the valve either OPEN / CLOSED / STOP.

There are three discrete valve control commands available:
1. 1 (0x0001) for Open
2. 2 (0x0002) for Close
3. 3 (0x0003) for Stop now at the current location.

The 1746-C will only react to these values.

The mechanism used to execute these commands is:
- If (OPEN) command: a setpoint of 4095 is transmitted to the actuator.
- If (CLOSE) command: a setpoint of 0 is transmitted to the actuator.
- If (STOP) command:
    - A Modbus command is sent to the actuator to read the current position.
    - That "current" position is sent to the actuator as the setpoint to go to.

A read back of Table [3] will
- If a **write** command issued: reflect the write command accepted by the 1746-C
- If a **read** command issued: reflect the last status of the discrete commands (unless changed by a new table [5] entry).

Note 1: *While the table will accept any value, any illegal data will be ignored.*

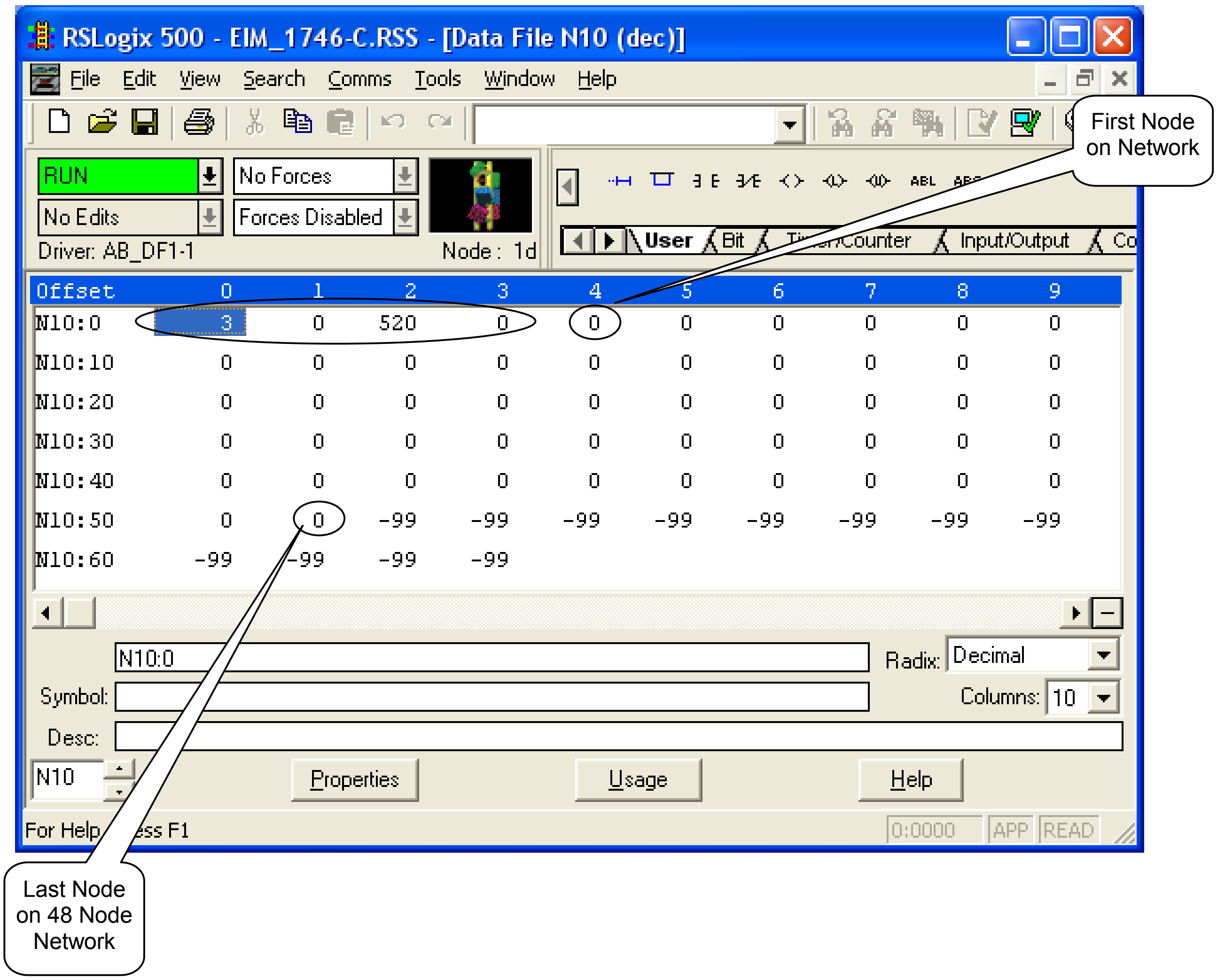
Note 2: *Any NEW discrete control command in this table will change the corresponding setpoint control value in Table [5] to either 0 / 4095 / or "current position".*

Note 3: *If a NEW change setpoint command is issued in Table [5] for a particular valve, then the corresponding discrete control command in this table (Table [3]) will be zeroed out.*

Example Table [3] Feedback (48-node network) …

Characteristics:

- There are no discrete commands waiting for execution.



5.4.4. Table [4]: Valve Position Indication … 0 – 100.0%

Words [4 → 63] PLC access: Read Only.

The Network Master receives the valve position from the actuator as an integer representing 0 → 100% open in 0.1% increments. In other words, it receives a value from 0 – 1000 and each increment indicates a 0.1%

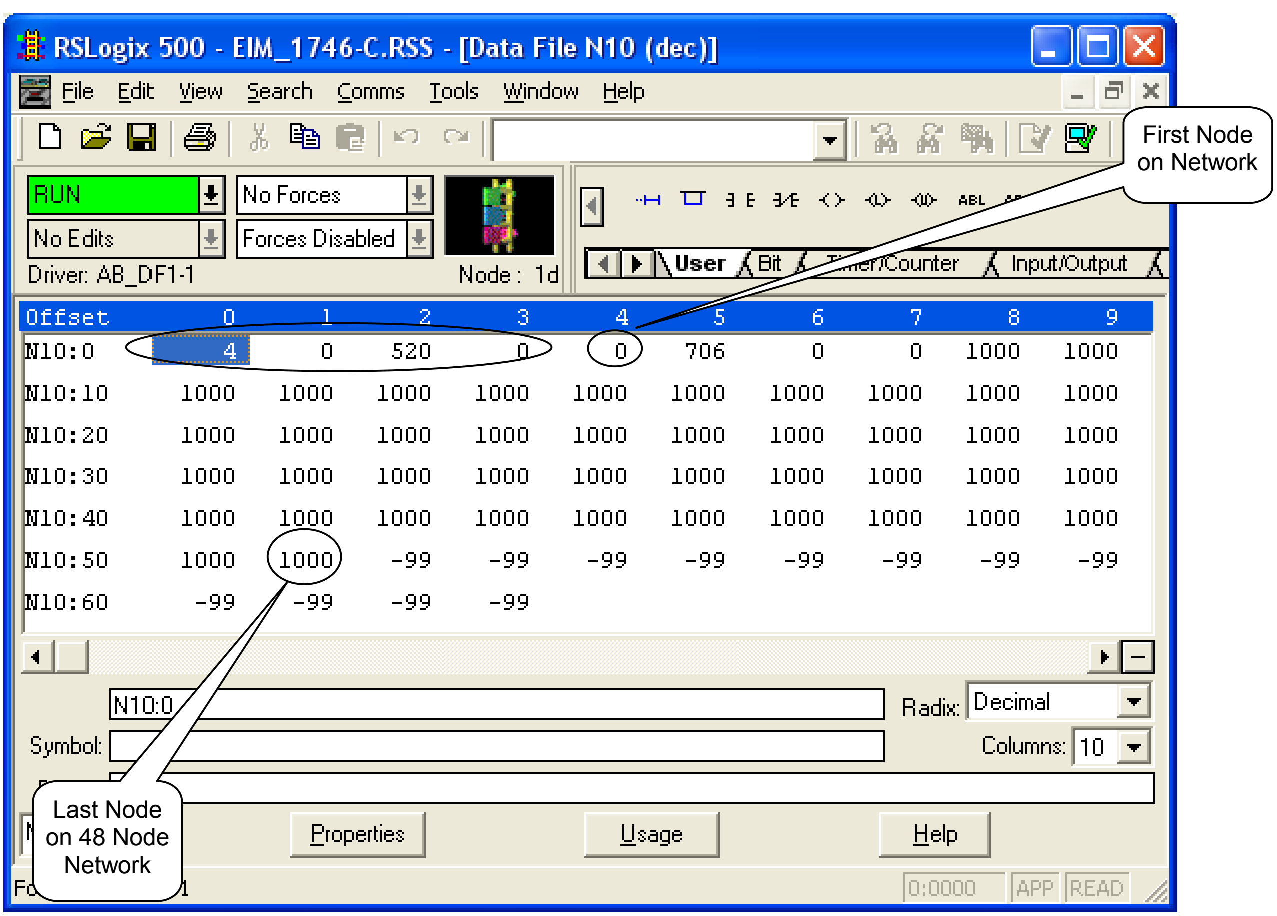
- 0 → Full Closed (0.0% open)
- 1000 → Full Open (100.0 % open)

This data is derived from holding register [13] in the actuator. It is retrieved with the "standard data request" - Modbus read command when the particular actuator is being polled (scanned).

Example Table [4] Feedback (48-node network) …

Characteristics:

- Units [1, 3, 4] indicated position is 0% open … probably not operating.
- Unit [2] indicated position is 70.6% open
- The rest of the network's indicated position is fully open at 100.0% open



5.4.5. Table [5]: Valve Position Setpoint … 0 – 4095

Words [4 → 63]   PLC access: **Read / Write**

This table commands the actuators to move to a setpoint position designated by the PLC. When the 1746-C sends the new value to the actuators (telling them to move), it uses the Modbus function 6 command to write to holding register [11] in the actuator.

By sending a numeric value from 0 → 4095, the actuator will be commanded to move to a position based on:

- 0 (0x0000) = 0% Open (closed)
- 4095 (0x0FFF) = 100% Open

A read back of Table [5] will

- If a **write** command issued: reflect the write command accepted by the 1746-C
- If a **read** command issued: reflect the current value of the setpoint register in the actuators.

Notes:

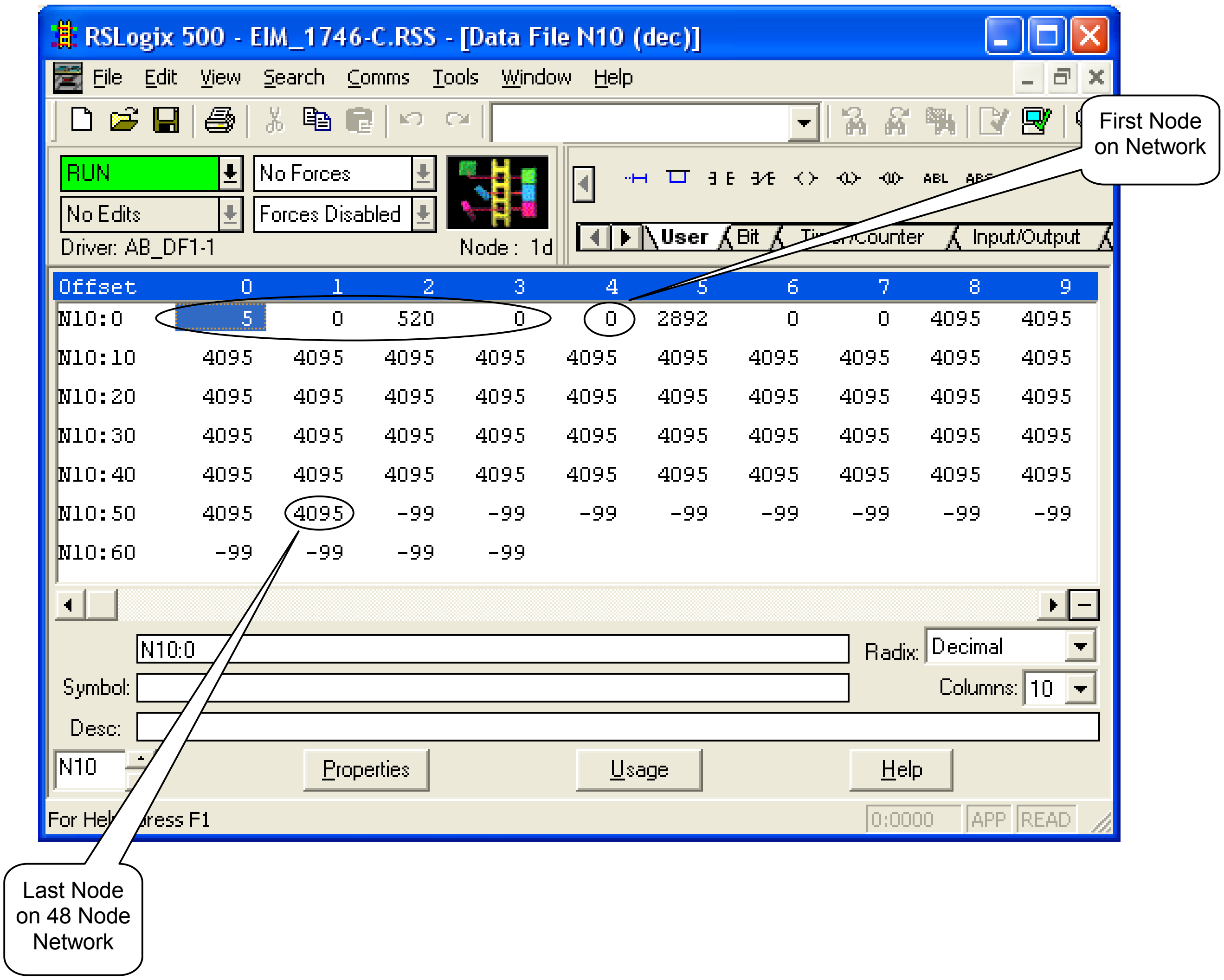
- *Only values 0 → 4095 are allowed to be written by the PLC … all others are ignored.*
- *Any EIM actuator may use position control via setpoint values. For the tightest position possible with minimal overshoot or undershoot, the actuator must have the solid-state or VFD control option. Please read the **Controlinc Quick Startup Guide** for more information.*
- *If the actuator position has been moved while the PLC is down or the valve was moved while in local control, then as soon as the PLC and actuators are communicating and accepting remote commands, it will move according to the existing value in this Table.*
- *Issuing a command in Table [5] will "zero out" any discrete (block) mode command issued on Table [3] for the particular valve.*
- *Any NEW discrete control command in table [3] will change the corresponding setpoint control value in Table [5] to either 0 / 4095 / or "current position"*
- *If the 1746-C ever returns a value of **0xFFFF** for a setpoint indication in this table, it means that the particular actuator is returning the standard "**invalid setpoin**t" and is currently stopped and waiting for a new command to move the valve.*

Example Table [5] Feedback (48-node network) …

Characteristics:

- Units [1, 3, 4] setpoint position is 0 (out of 4095) … 0% open … either fully closed or not operating.

- Unit [2] setpoint position is 2892 (out of 4095) … ~~ 70.6% open
- The rest of the network's setpoint position is fully open … 4095 out of 4095



5.4.6. Tables [6 & 7]: User Analog Inputs #1 & 2

Words [4 → 63]   PLC access: Read Only.

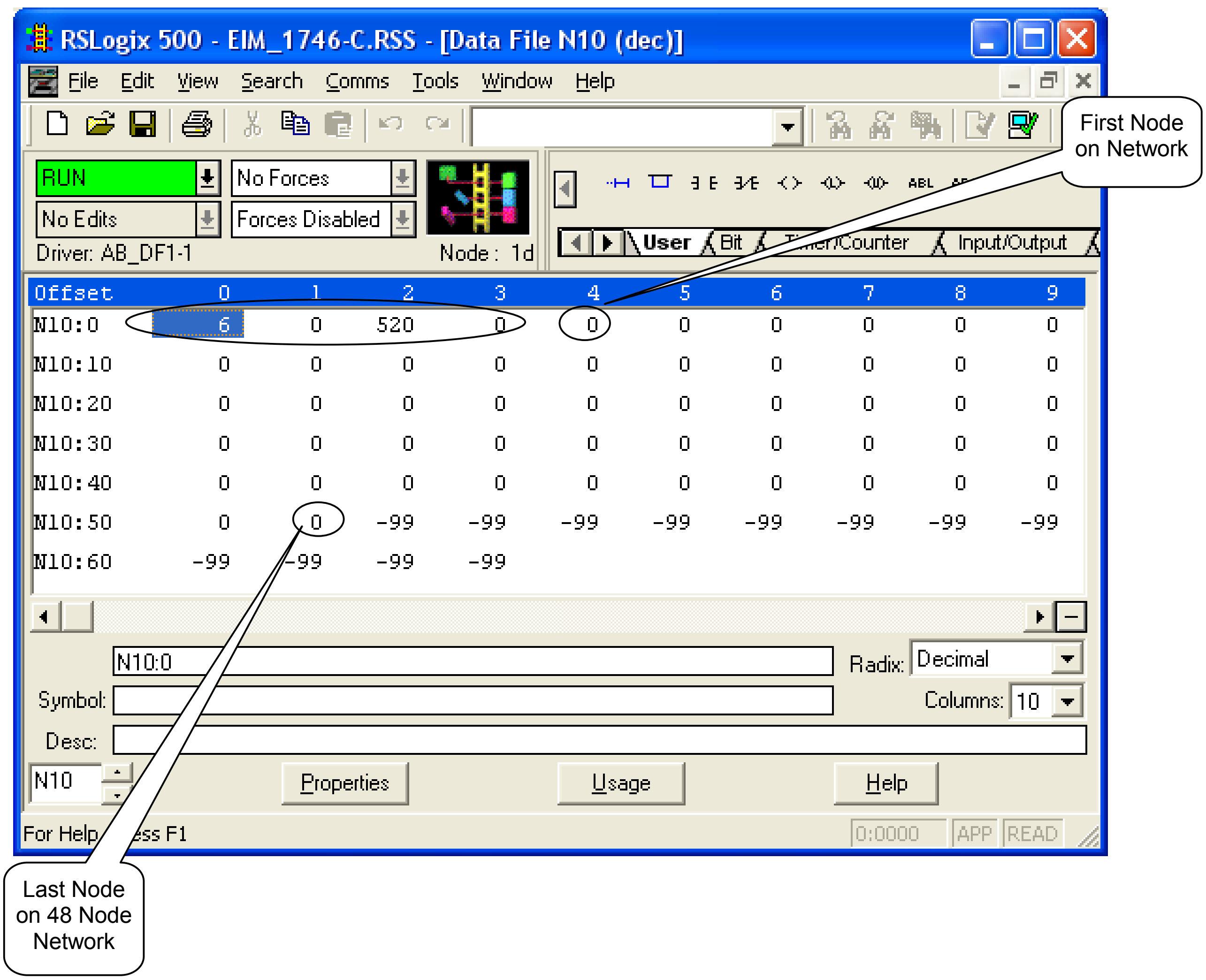
These tables contain 12-bit numbers (0 → 4095) representing the current value for analog input channels. This data is derived from holding registers [16 & 17] in the actuators and is retrieved with the "standard data request" - Modbus read command when the particular actuator is being polled (scanned).

- Table [6]: Value for User Analog Input #1 (holding register [16] in the actuators)
- Table [7]: Value for User Analog Input #2 (holding register [17] in the actuators)

Example Table [6] Feedback (48-node network) …

Characteristics:

- There are no values on User Analog Input #1 on any unit on the network … all zero.



5.4.7. Table [8]: Analog Output #1

Words [4 → 63] PLC access: **Read / Write**

This table/command tells the actuator to place an appropriate 4-20 mA signal on its analog output channel. This data is derived from holding register [10] in the actuators and is retrieved with the "standard data request" - Modbus read command when the particular actuator is being polled (scanned).

When the 1746-C sends a new value to the actuators, it uses the Modbus function 6 command to write to holding register [10] in the actuators.

The output is a linear scaling based on:

- 0 (0x0000) = 4 mA
- 4095 (0x0FFF) = 20 mA

Note: *Only values 0 → 4095 are allowed … all others are ignored.*

A read back of Table [8] will

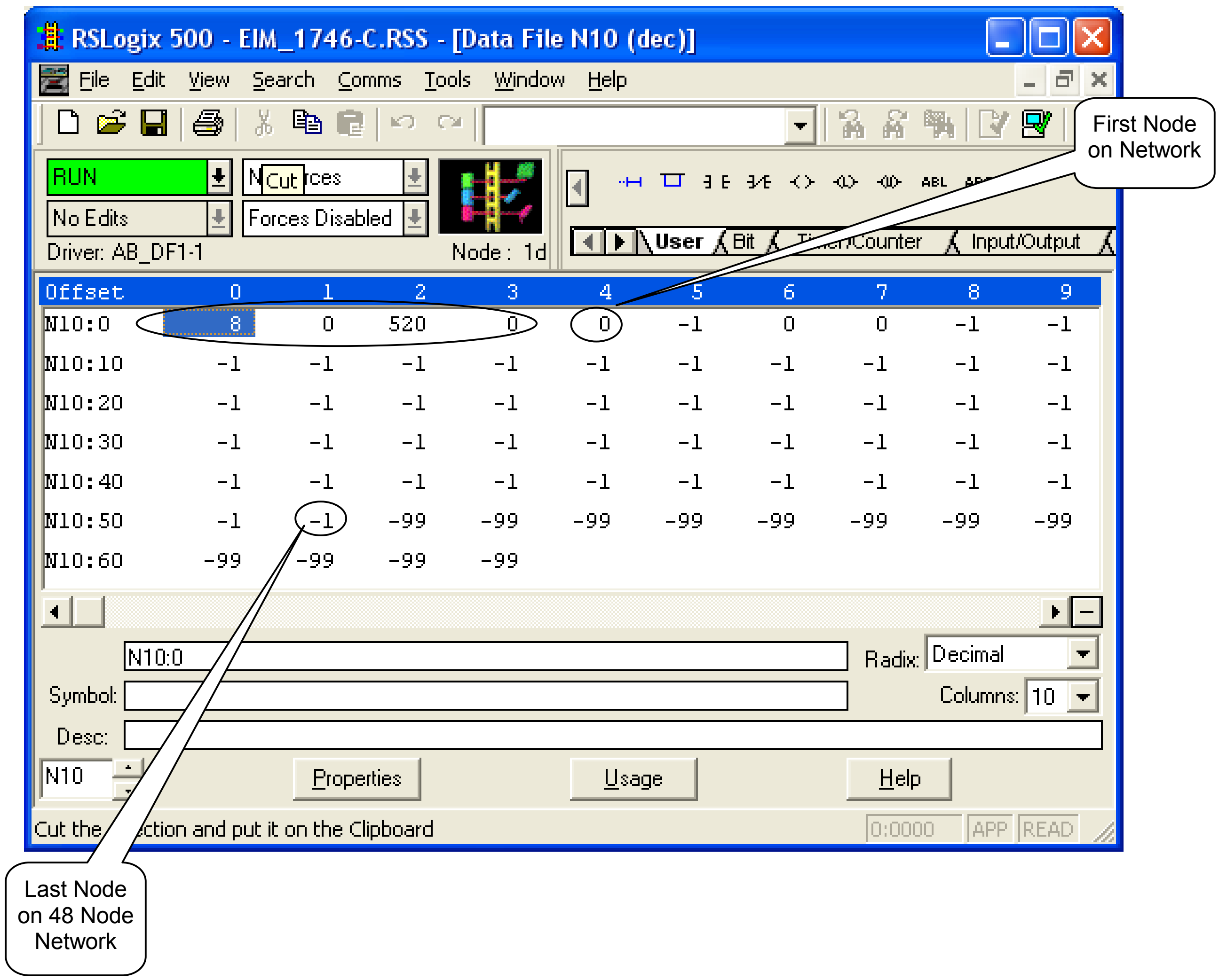
- If a **write** command issued: reflect the write command accepted by the 1746-C
- If a **read** command issued: reflect the current value of the analog output register in the actuators.

Note: *Please read the Operators Manual the particular actuator for more information.*
Note: *This is an option that must be installed on the actuator.*

Example Table [8] Feedback (48-node network) …

Characteristics:

- On this network, none of the analog output values have been activated … a "-1" value recorded there (valid values are 0 – 4095).



5.4.8. Tables [9 & 10]: Digital Input Accumulators (Totalizers) #1 & 2

Words [4 → 63]   PLC access: **Read / Write**

These tables contain 16-bit accumulator (totalizer) numbers stored in holding registers [66 & 67] in the actuators. This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

- Table [9]: Value for Accumulator (Totalizer) #1
- Table [10]: Value for Accumulator (Totalizer) #2

These tables are not polled for unless the 1746-C is configured to allow it …

- if Table [0] – Word [7] == 0: The accumulator/totalizer information is NOT requested when actuators are polled (*the default*).

- if Table [0] – Word [7] =/= 0: The accumulator/totalizer information is requested when actuators are polled.

The PLC may reset these tables (and the counters in the actuators) with any value is prefers. All it needs to do is insert the new value in the table and send it to the 1746-C as a write command. The 1746-C will then send the value to the actuator.

A read back of Tables [9 & 10] will

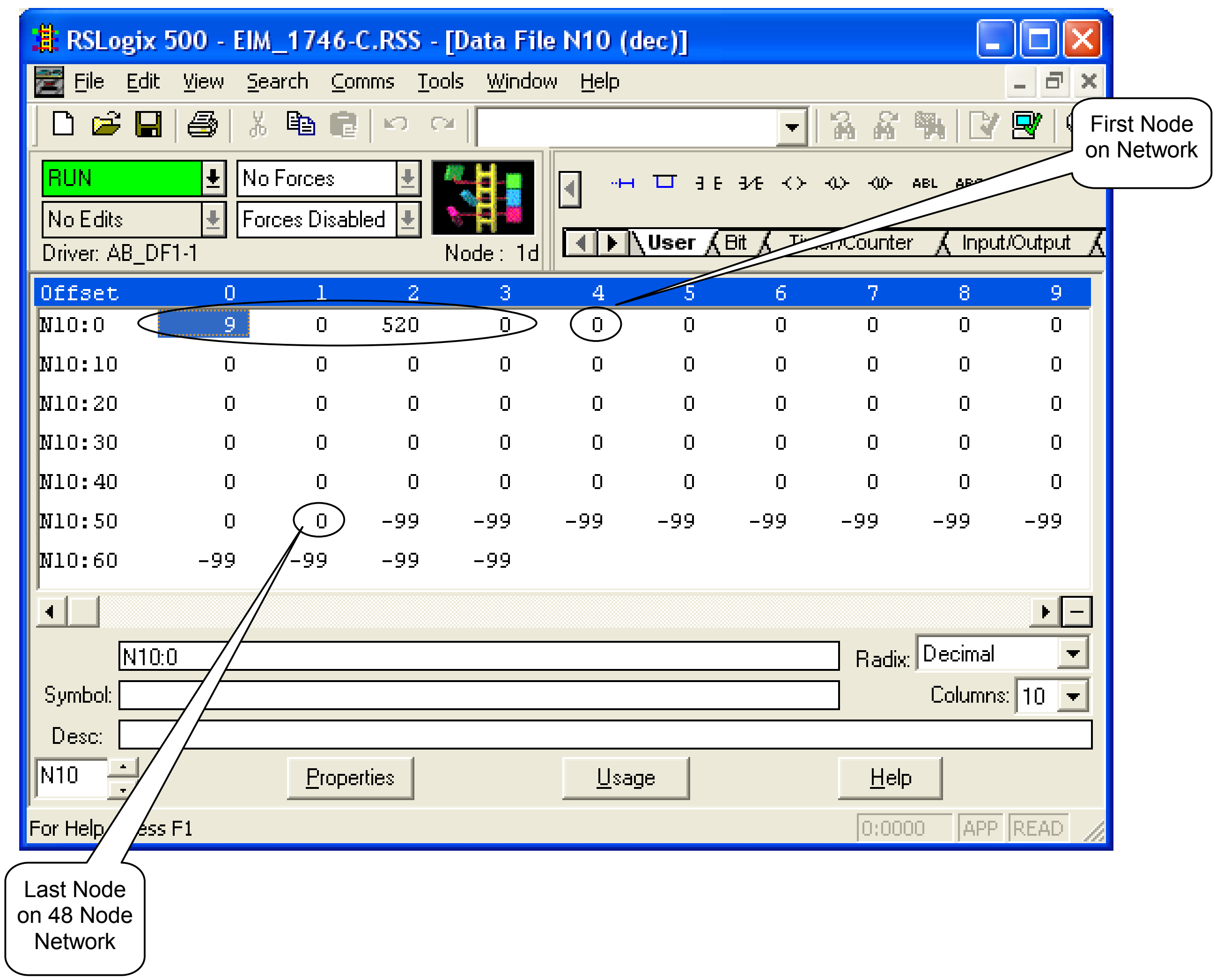
- If a **write** command issued: reflect the write command accepted by the 1746-C
- If a **read** command issued: reflect the last known values of the totalizer registers in the actuators.

Example Table [9] Feedback (48-node network) …

Characteristics:

- Totalizer #1 has not engaged for any unit on the network. This means that the digital input(s) assigned to this totalizer/accumulator have not toggled low/high since the last power up on any unit on the network.
    - On 320A/B systems … user input #1.
    - On TEC2000 systems … it depends upon the configuration.

- This also means that the feature is probably not implemented on the actuators.



5.4.9. Table [11]: Valve Position Indication … 0 – 4095

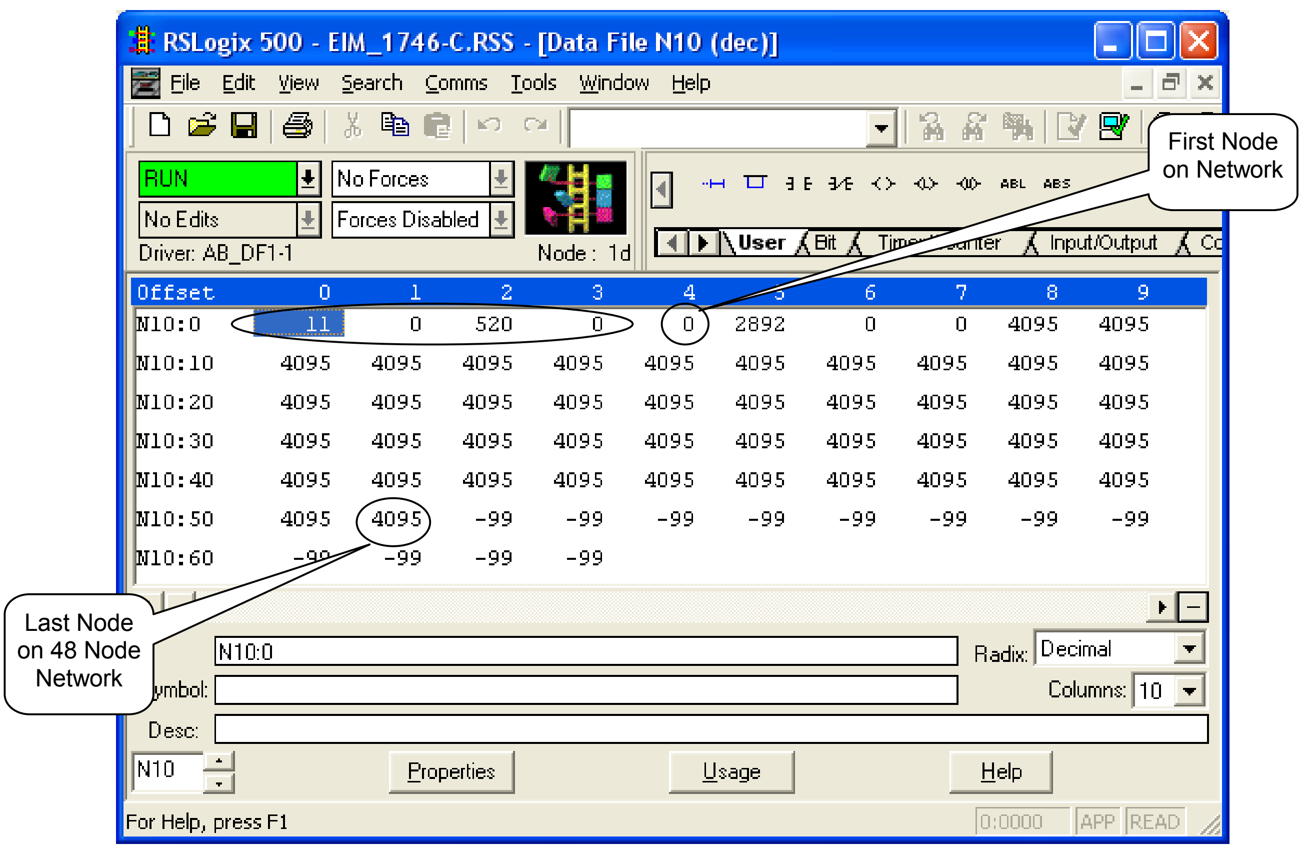
Words [4 → 63]   PLC access: Read Only.

This table contains the 0-4095 representation of the valve position for the actuators. This data is derived from holding register [14] in the actuators and is retrieved with the "standard data request" - Modbus read command when the particular actuator is being polled (scanned).

Example Table [11] Feedback (48-node network) …

Characteristics:

- Units [1, 3, 4] position indication is 0 (out of 4095) … 0% open … either fully closed or not operating.
- Unit [2] position indication is 2892 (out of 4095) … ~~ 70.6% open
- The rest of the network's position indication is fully open … 4095 out of 4095



5.4.10. Table [12]: Discrete Input Statuses

Words [4 → 63] PLC access: Read Only.

This table contains the status indications of the discrete inputs on 320A/B systems. Each bit in each word in the table indicates the status of a particular discrete (hardwired) digital input for a particular actuator. This data is derived from holding register [05] in the actuators and is retrieved with the "standard data request" - Modbus read command when the particular actuator is being polled (scanned).

If a TEC2000 was polled, the TEC2000 system correlates appropriate values to the bits being returned.

This status word can be "bit parsed" to give:

- Bit-0: LSO (Limit Switch – Open) switch input is active (is tripped)
- Bit-1: LSC (Limit Switch – Close) switch input is active (is tripped)
- Bit-2: AUX Open Contactor input is active (is engaged)
- Bit-3: AUX Close Contactor input is active (is engaged)

- Bit-4: SS in Local Position input is active (is tripped)
- Bit-5: SS in Remote Position input is active (is tripped)
- Bit-6: TSO (Torque Switch – Open) input is active (is tripped) … not the alarm!
- Bit-7: TSC (Torque Switch – Close) input is active (is tripped) … not the alarm!

- Bit-8: Power Monitor Alarm input is active.
- Bit-9: Thermal Overload Alarm input is active
- Bit-10: Phase Monitor Alarm input is active (only if Phase Sentry is Installed)
- Bit-11: Local (Hardwired) Alarm input is active

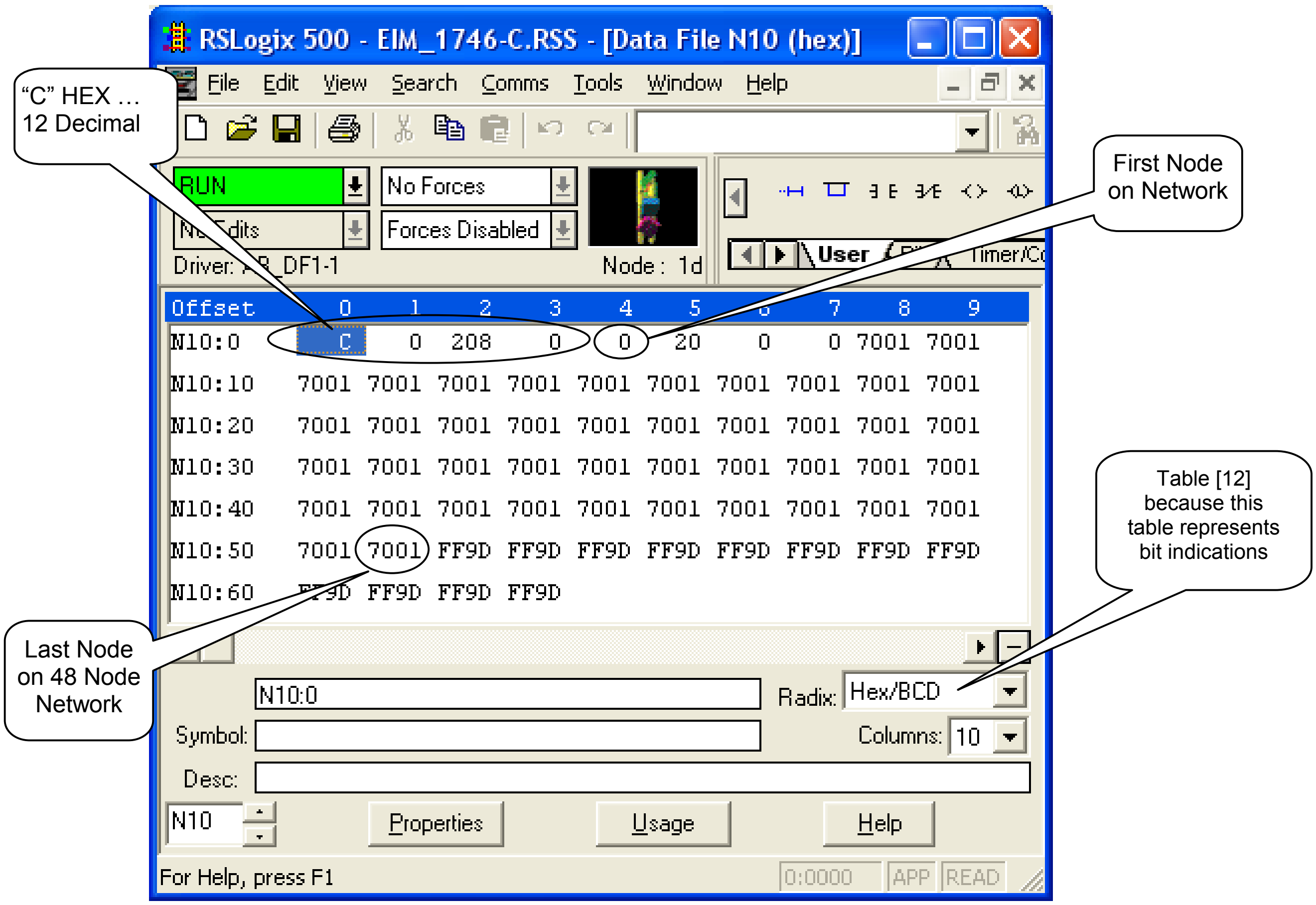
- Bit-12: AUX Alarm input is active
- Bit-13: User Input #1 is active
- Bit-14: User Input #2 is active

- Bit-15: Reserved … no operation.

Note: Bit-0 is the LSB in the word. Bit-15 is the MSB in the word.

Some of these bit indications are repeats of those in table [2]. However, not all.

Example Table [12] Feedback (48-node network) …



5.4.11. Table [13]: Solid State Relay (SSR) Configuration Table

Words [4 → 63] PLC access: **Read / Write**

This is a read/write table that is used to help configure the actuators on the network. It is intended to work with actuators that are setup for modulating control mode. If a 320A actuator has an **OPTIONAL** SSR installed the network master must ensure the actuator knows it is configured with an SSR.

By writing a non-zero **POSITIVE** value to these positions in table [13], tells the network master to tell the particular 320A actuator to configure itself to use an SSR … set coil [37] = 1 on the actuator. If not a 320A actuator, this table value has no purpose.

SUMMARY:

- If word [n] == 0 (where "n" is 4 → 63) … this actuator does not have a SSR.
- If word [n] > 0 (any non-zero **POSITIVE** value) … this actuator has a SSR.

An example would be a network of actuators in which 3 are 320A systems that have SSCs … lets say on valves 2, 4 and 10 by Modbus address. The corresponding words for these actuators are:

- Valve 2: Word [5] … (2 + 3)
- Valve 4: Word [7] … (4 + 3)
- Valve 10: Word [13] … (10 + 3)

A read back of Table [13] will verify the write command being accepted by the 1746-C.

Note: *If an actuator*

- *has an SSR installed AND*
- *it is NOT configured for SSR control AND*
- *it is placed in modulating mode*

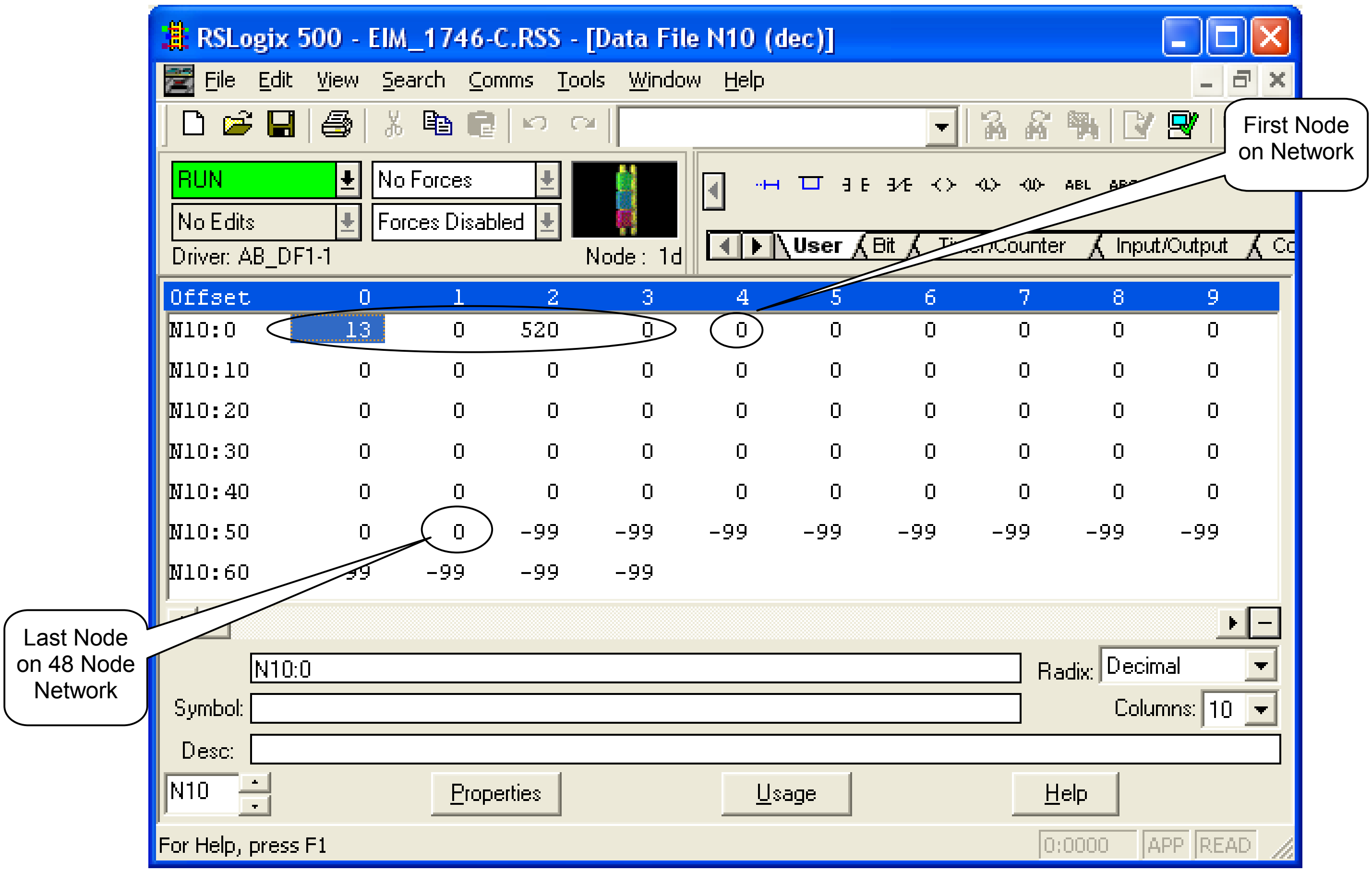
then the remote host will be unable to control the valve via setpoint control. The actuator will read the desired setpoint positions but will not be able to send the valve to the position and effectively stop it there.

Note*: Remember, setting this configuration bit is only necessary if the unit is a 320A system … 320B and TEC2000 systems do not need this configuration assistance.*

Example Table [13] Feedback (48-node network) …

Characteristics:

- None of the units are configured for an SSR … probably either because they just don't have one or because they are not 320A systems.



5.4.12. Table [14]: Additional Register Being Polled From Entire Network

Words [4 → 63] PLC access: **Read / Write**

This operation allows the operator the chance to poll the entire network for any particular holding register. The register values polled are stored in this table. This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

This table is not polled for unless the 1746-C is configured to allow it …

- Table [0] – Word [10] … contains the register number to poll for.

- Table [0] – Word [10] … is a flag that indicates whether or not to start requesting that register from the actuators when scanning such that …

    - if Table [0] – Word [11] == 0: The register is not polled (*the default*).

    - if Table [0] – Word [11] =/= 0: The register information is requested when actuators are polled.

The PLC may reset this table with any value is prefers. To do this, the PLC must insert the new value into the table and send it to the 1746-C as a write command. The values in this table are not sent out to the actuators on the network.

A read back of Table [14] will
- If a **write** command issued: reflect the write command accepted by the 1746-C
- If a **read** command issued: reflect the last known values of the specified register polled in the actuators.

If the PLC is utilizing table [14], it is recommended that the PLC also poll table [20]. If an exception message is returned, then the value won't be inserted into table [14] but the exception message indicator will be inserted into table [20].

If detected, and the PLC wants to try again, it should initialize table [14] to some known value and either explicitly clear table [20] or wait until after the next diagnostic scan (which automatically clears table [20]) before trying again.

Example Scenario: If there are multiple types of EIM equipment on the network and the table [14] command requests data from an actuator that is "not supported by that type of actuator" (but is on the others), the actuator will return a Modbus Exception for that message and the values will NOT be loaded into tables [14 or 15]. However, the exception codes will be loaded into table [20].

For example, if requesting register [1010] from a
- 320A system … an exception code is returned.
- TEC2000 system … the register containing coils [1069 – 1111] is returned.

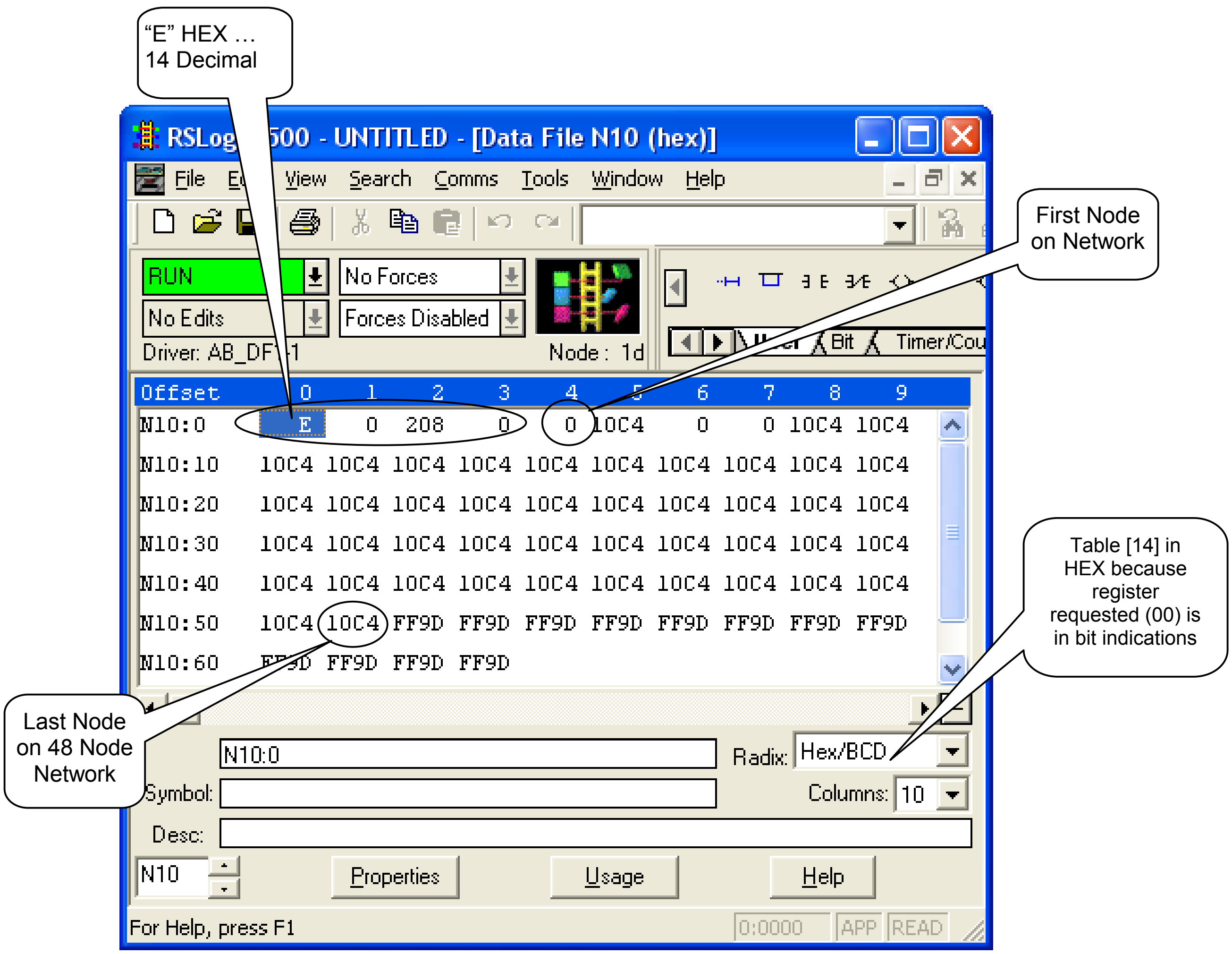
Example Table [14] Feedback (48-node network) …

Characteristics:

- (register [00] read back from the entire network … cannot tell from here … in table [0])

- To make it more meaningful, table [14] is in hex because register [00] is bit indications.

- Units [1,3,4] have a value of 0 … coils [0-15] all off … units probably off line.
- Unit [2] has a value of 0x10C4 … Coils set: [12, 7, 6, 2]
- Units [5-48] have a value of 0x10C4 … Coils set: [12, 7, 6, 2]



5.4.13. Table [15]: Additional Block of Registers Being Polled From 1 Actuator

Words [4 → 63] PLC access: **Read / Write**

TABLE FORMAT:
This operation allows the operator the chance to poll a single actuator for a block of registers. The register values returned are stored in table [15] such that …

- Words [4 - 7] … header information about information returned …
  - Word [4] … Valve address where block of data came from.
  - Word [5] … Valve (actuator) type id.
  - Word [6] … Starting holding register number.
  - Word [7] … Total Number of registers in block.

- Words [8 - 13] … not used … reserved.
- Words [14 - 63] … register information returned.

TABLE CONTROL:
Control of this operation is via 4 words in table [0] such that …

- Table [0] – Word [11] … The actuator address to get the block of registers from.
- Table [0] – Word [12] … The register number for the 1st holding register in the block to retrieve.
- Table [0] – Word [13] … The total number of registers to retrieve.
- Table [0] – Word [14] … This is a flag that indicates whether or not to start requesting that block of registers from that actuator when polling it.

If Table [0] – Word [14] == 0 do not poll for the block of registers.
If Table [0] – Word [14] =/= 0 (any non-zero value) … poll for the block of registers.

By default, Table [0] – Words [11 → 14] == 0 … do not poll.

OPERATIONAL OVERVIEW:
This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

The PLC may reset this table with any value is prefers. To do this, the PLC must insert the new value into the table and send it to the 1746-C as a write command. The values in this table are not sent out to the actuators on the network.
A read back of Table [15] will

- If a **write** command issued: reflect the write command accepted by the 1746-C except words [4 – 13] are zeroed out.

- If a **read** command issued: reflect the last known values in the table.

Words [4 – 7] are only filled with non-zero values if the block of data in words [14 – 63] table are actually values retrieved from the actuator at the address specified in word [4].
Therefore, if a read of table [15] indicates that

- Word [4] == 0, then the data is not valid (there are not valves with address 0 that can respond with data).
- Word [4] =/= 0, then the data in the table came from a valve.

If the PLC is utilizing table [15], it is recommended that the PLC also poll table [20]. If an exception message is returned, then the block of data won't be inserted into table [15] but the exception message indicator will be inserted into table [20].

If detected, and the PLC wants to try again, it should initialize table [15] to some known value and either explicitly clear table [20] or wait until after the next diagnostic scan (which automatically clears table [20]) before trying again.

Example Scenario: If there are multiple types of EIM equipment on the network and the table [15] command requests a block of data from an actuator that is "not supported by that type of actuator" (but is on the others), the particular actuator will return a Modbus Exception for that message and the block of data will NOT be loaded into tables [15]. However, the exception codes will be loaded into table [20].

For example, if requesting a block of registers [15 - 30] from unit #5 and it is a

- 320A system … the block of registers is returned.
- TEC2000 system … an exception code is returned.

Notes on table [15] use:

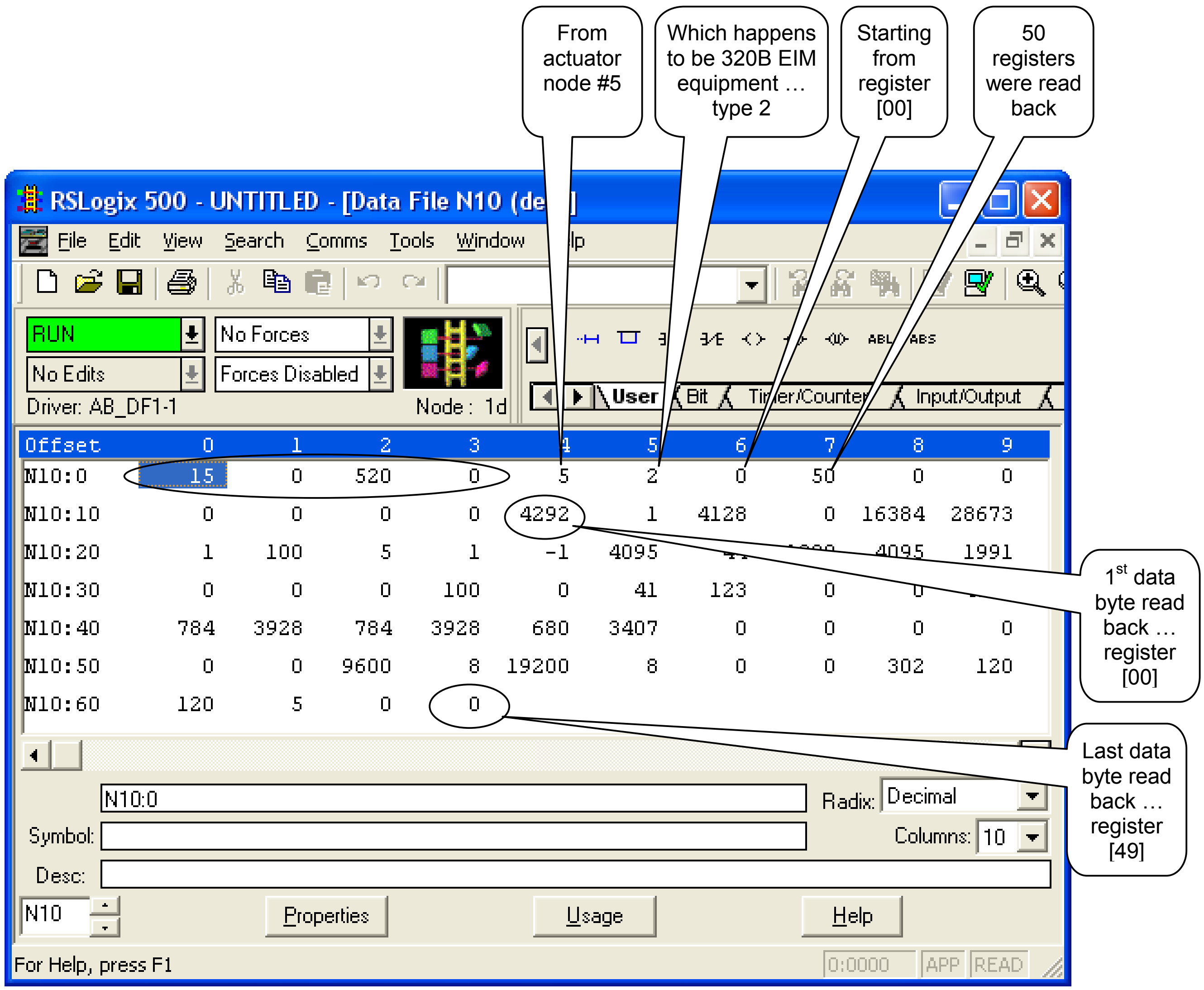
- It is recommended that the PLC only keep this operation active until the data is received and then deactivate it (setting table [0] – word [14] = 0).

- When the PLC writes data to this table, the 1746-C only overwrites the last 50 words with the data sent in. The rest of the table is zeroed out.

Example Table [15] Feedback (48-node network) …

Characteristics …

- From actuator node #5
- Which happens to be 320B EIM equipment … type 2
- Starting from register [00] and
- 50 registers were read back.



5.4.14. Tables [16 & 17]: Monitor & Control Discrete Digital Outputs

Words [4 → 63] PLC access: **Read / Write**

The entire operation to monitor and control the discrete outputs requires 2 tables ...

- Table [16] ... CONTROL --- store commands (desired states) for discrete outputs (coils).
- Table [17] ... MONITORING --- feedback (actual states) of discrete outputs (coils).

Table [16] allows the operator the ability to monitor (poll) and control the discrete outputs on an actuator. All discrete output control is performed via this table (no table [0] entries are required).

Table [17] is the feedback information for polling the discrete outputs.

- This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).
- This data is only polled for when the matching valve entry in table [16] indicates that the 1746-C should start monitoring these coils. Once polling (monitoring) has started, it must be explicitly turned off.
- This table will only reflect the status of bits representing coils. All others are either ignored or zeroed out.
- The PLC may reset this table with any value is prefers. All it needs to do is insert the new value in the table and send it to the 1746-C as a write command. The table will be reset even though nothing will be sent out to any actuator.

TEC2000 DISCRETE OUTPUTS (COILS)

On a TEC2000 system, there are 12 outputs and they are addressed as coils [1000 - 1011] in register [1004] such that …

- Coil [1000] … RO1 … Relay Output # 1
- Coil [1001] … RO2 … Relay Output # 2
- Coil [1002] … RO3 … Relay Output # 3
- Coil [1003] … RO4 … Relay Output # 4
- Coil [1004] … RO9 … Relay Output # 9
- Coil [1005] … RO10 … Relay Output # 10.
- Coil [1006] … RO11 … Relay Output # 11
- Coil [1007] … RO12 … Relay Output # 12.
- Coil [1008] … SO6 … Solid State Output # 6
- Coil [1009] … SO7 … Solid State Output # 7
- Coil [1010] … SO8 … Solid State Output # 8
- Coil [1011] … RO9 … Relay Output # 9

Ordering is such that

- Coil [1000] is the LSB and located in the “bit-0” position … the right most position.
- Coil [1011] is the MSB and located in the “bit-11” position.

320A/B DISCRETE OUTPUTS (COILS)

On a the other system types (320A and 320B), there are 3 outputs and they are addressed as coils [3-5] in register [00] such that …

- Coil [3] … ESD / monitor relay.
- Coil [4] … User Output Relay #1
- Coil [5] … User Output Relay #2

Ordering is such that

- Coil [3] is the “bit-3” position … the 4th position in from the right.
- Coil [4] is the “bit-4” position … the 5th position in from the right.
- Coil [5] is the “bit-5” position … the 6th position in from the right.

FORMAT

There are 2 distinct areas in each word in table [16] …

- Command / Control bits … bits [15-13] in the upper most nibble.
- Desired coil (output) states …
  - Bits [11 – 0] for TEC2000 systems
  - Bits [5-3] in other systems.

Only these bits are set in this table. All others are zeroed out.

COMMAND / CONTROL

There are only 3 valid control values (upper nibble in the word) the PLC can send to the 1746-C concerning this operation …

1. **1000 (binary)**: Setting the upper nibble = 1000 … (top 1 bit set) tells the 1746-C to start monitoring the status of the output coils and recording them in table [17]. The 1746-C will also clear the desired state bits … set table [16] – bits [3-5] = 000.

2. **1110 (binary)**: Setting the upper nibble = 1110 … (top 3 bits set) tells the 1746-C to
   - Start monitoring the status of the output coils and recording their states in table [17].
   - ONE TIME … write the desired states of the output coils to the particular actuator. Once successful, bit [13] is set = 0 to prevent further (excess) writes.

3. **0xxx (binary)**: Setting the bit = 0 … (bit [15] = 0 and don’t care about the others) tells the 1746-C to terminate monitoring and writing. Essentially, clear the command. The 1746-C will also clear the bits representing the desired state of all output coils in table [16].

The 1746-C will only react to these values from the PLC in the upper nibble. All others are ignored.

DESIRED STATES

The desired states of the digital outputs are directly reflected in the bits in the table … which directly reflect the coils (bits) in the holding registers in the actuator. This means that the bits still maintain their original position in all the registers …

For example,
- Coil [3] … is in register [00] – bit [3] … and in tables [16 & 17] in bit position [3]
- Coil [4] … is in register [00] – bit [4] … and in tables [16 & 17] in bit position [4]
- Coil [5] … is in register [00] – bit [5] … and in tables [16 & 17] in bit position [5]

Therefore, the PLC must set/clear bits [3-5] to control the outputs.

For example,
- Coil [1000] … is in register [1004] – bit [0] … and in tables [16 & 17] in bit position [0]
- Coil [1005] … is in register [1004] – bit [5] … and in tables [16 & 17] in bit position [5]

Therefore, the PLC must set/clear bits [0,1,5] to control these outputs.

IMPORTANT NOTE

Since a single command is sent to the actuator that affects all discrete outputs on a system at one time, it is important that the PLC be aware of the "state" of all coils on the particular valve before sending a command because ALL OUTPUTS ON THE VALVE will be affected EVERYTIME one of these commands is sent. This is why
- Monitoring can be activated without writing values.
- Monitoring remains active after a write until explicitly turned off.

By monitoring table [17], the PLC can ensure correct behavior.

A read back of Table [16] will
- If a **write** command issued: reflect the status of the network write & network monitor commands accepted by the 1746-C.
- If a **read** command issued: reflect the last status of the network write & network monitor commands issued out to the network.

Notes on table [16] use:
1. The PLC can monitor when a write happened by viewing the upper nibble of the word for the particular valve. When it changes from (1110 … 0xE) →→ (1100 … 0xC) the write has happened and the system is still monitoring the states of the relays.
2. The PLC can activate/deactivate monitoring these relays any time by setting/clearing bit [15] (*see the "Command / Control" section above*).
   However, anytime the PLC writes a value to the coils in an actuator, monitoring is automatically started … and will not terminate until the PLC terminates it.
   Therefore, the PLC needs to monitor this and deactivate the operation when not needed anymore … otherwise, the additional polling will slow (lengthen) network scan times.
3. The 1746-C uses a Modbus Command 05 to write multiple coils when writing values to the actuators.

Example Tables [16 & 17] Feedback (48-node network) …

Characteristics:

- To make it more meaningful, tables [16 & 17] are in hex because of bit indications.

- Monitoring is being performed on unit [1]
  (bit [15] = 1 in table [16])

- The PLC commanded the 1746-C to set all the coils on units [2 – 6]
  (bits [15-13] = 111 … 0xE0xx in table [16] … the value prior to the 1746-C acting upon the command)
  (bits [5-3] = 111 in table [16])

- The 1746-C processed the command and wrote the new coil states to actuators [2-6].
  (bits [15-13] = 110 … 0xC0xx in table [16])

- Units [2-6] all have all 3 discrete outputs on.
  (bits [5-3] = 111 … 0x0038 in table [17])

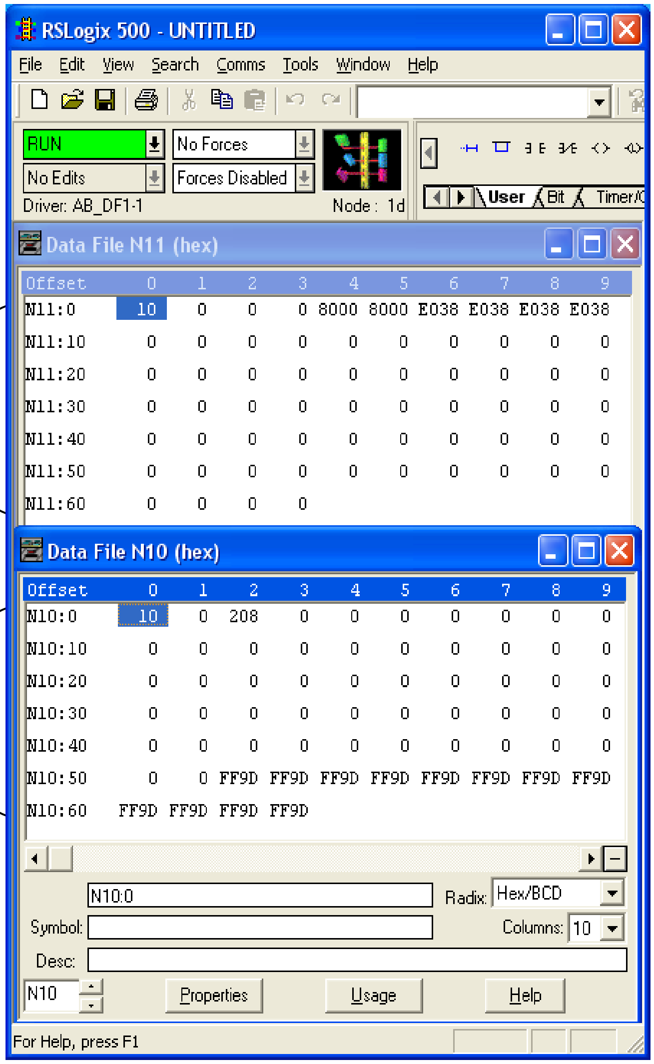
TABLE EXAMPLES ARE ON NEXT PAGE …

Tables [16 & 17] … with commands loaded but not sent

Tables [16 & 17] … after commands sent

Table [16] Sent out

Table [16] Feed back



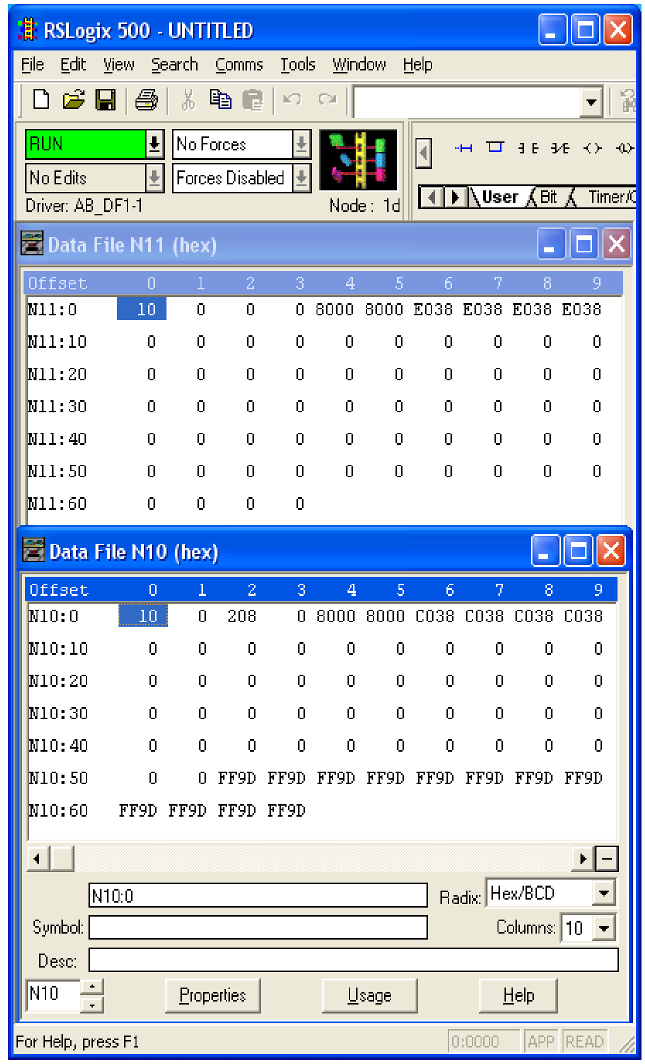
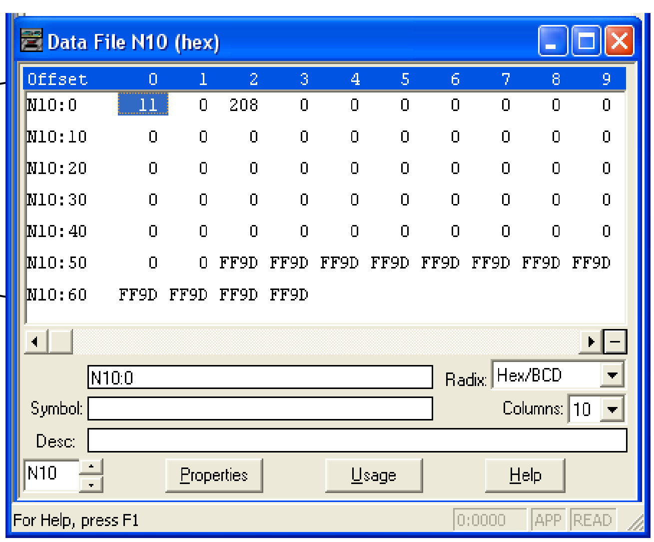
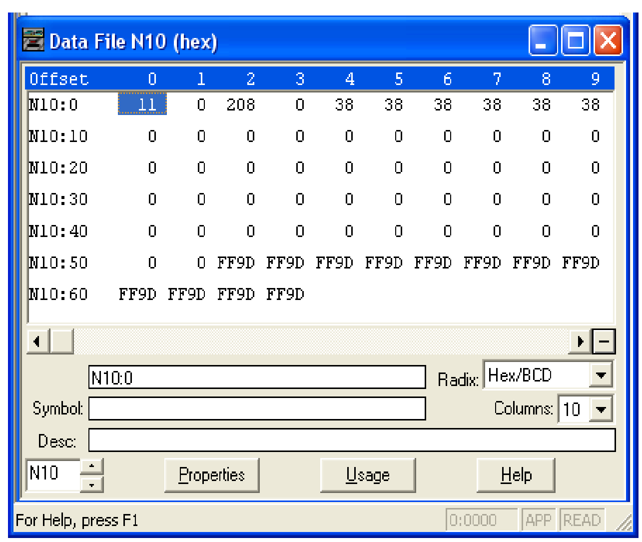


Table [17] Feed back





5.4.15. Table [18]: Actuator System Type ID

Words [4 → 63]   PLC access: Read Only.

This data is primarily derived from holding registers [44] & [100] in the actuator. These registers are retrieved with 2 separate Modbus read commands when the particular actuator is being polled (scanned) during a diagnostic scan.

There are currently 4 types of EIM equipment identified. There are 6 system type values that can be entered into table [18]:

- If EIM equipment type = 320A …   value in table [18] = “1”
- If EIM equipment type = 320B …   value in table [18] = “2”
- If EIM equipment type = TEC2000 …   value in table [18] = “3”

- If non-EIM equipment type …   value in table [18] = “0” or “255”

NOTE: There are 2 possible values if the unit is identified as non-EIM:

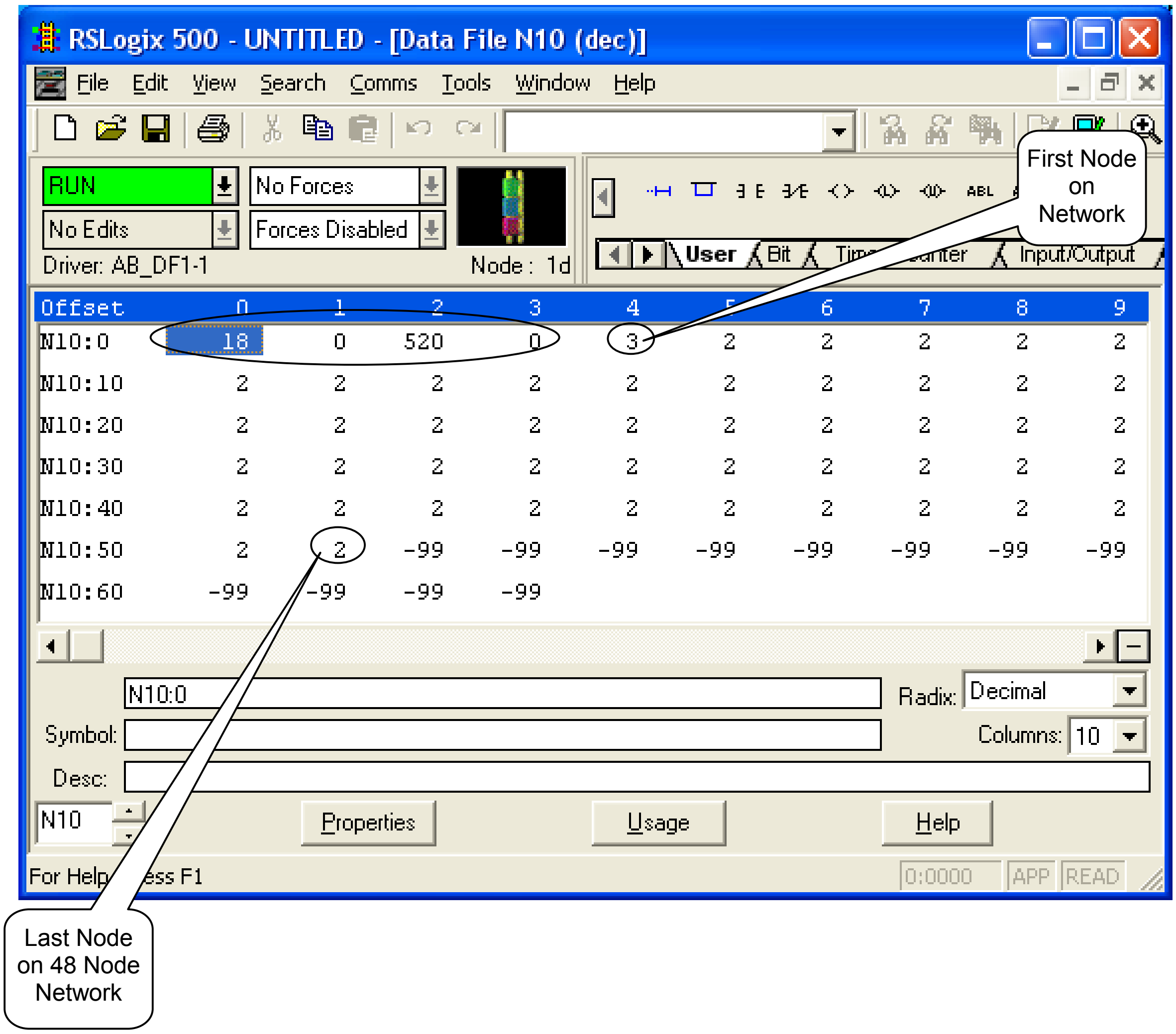
- 0 … If failed identification … method 1.
- 255 … If failed identification … method 2.

The 1746-C uses these values internally.

Example Table [18] Feedback (48-node network) …

Characteristics:

- Unit [1] is an EIM TEC2000 system … type 3
- Units [2 – 48] are EIM 320B systems ... type 2



5.4.16. Table [19]: Actuator Firmware Version ID

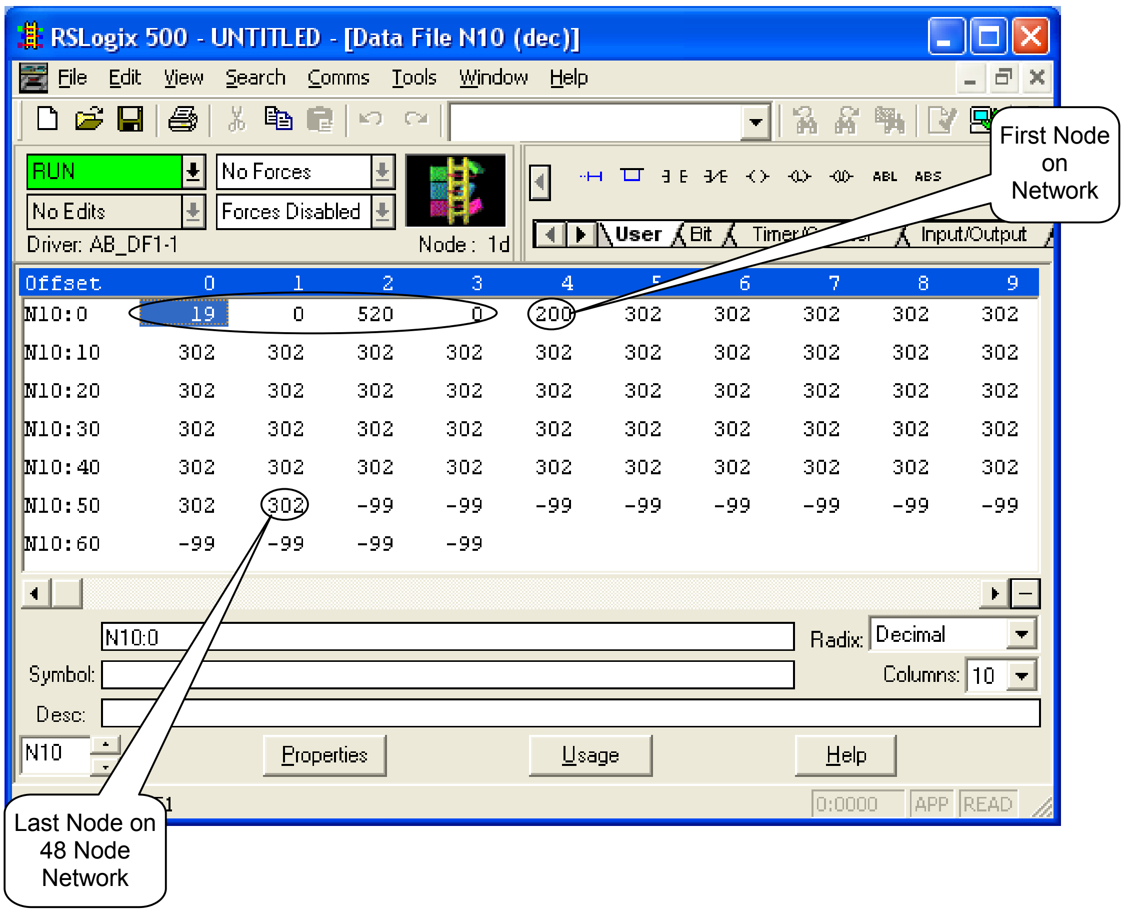
Words [4 → 63] PLC access: Read Only.

This data is derived from holding register [44] in the actuator. It is retrieved during a diagnostic scan when the system type ID is being determined. This table is filled the same time table [18] is filled. For more information, see table [18].

Example Table [19] Feedback (48-node network) …

Characteristics:

- Unit [1] is an EIM 320A system … firmware version: 2.00
- Units [2 – 48] are EIM 320B systems … firmware version: 3.02



Note: *TEC2000 systems do not store their version ID in register [44]. What you will see from a TEC2000 system is the upper bit set to "1" and the rest zero … 0x8000 hex (-32768 decimal). For more information relating to TEC2000 version identification, see the TEC2000 manual.*

5.4.17. Table [20]: Modbus Exception Message Response

Words [4 → 63] PLC access: **Read / Write**

This data is filled anytime a communication error is detected due to the remote system sending a 5-byte Modbus Exception Response message. These are "High-Level" Communication Errors … Application Level. This table is cleared at the start of a diagnostic scan.
Format:

- Table [20] - Hi-byte: Error'd function code as returned by the remote system.
- Table [20] - Lo-byte: Modbus Exception Type (1 → 3)

The PLC may reset this table with any value is prefers. To do this, the PLC must insert the new value into the table and send it to the 1746-C as a write command. The values in this table are not sent out to the actuators on the network.

A read back of Table [20] will

- If a **write** command issued: reflect the write command accepted by the 1746-C
- If a **read** command issued: reflect the current values in the table.

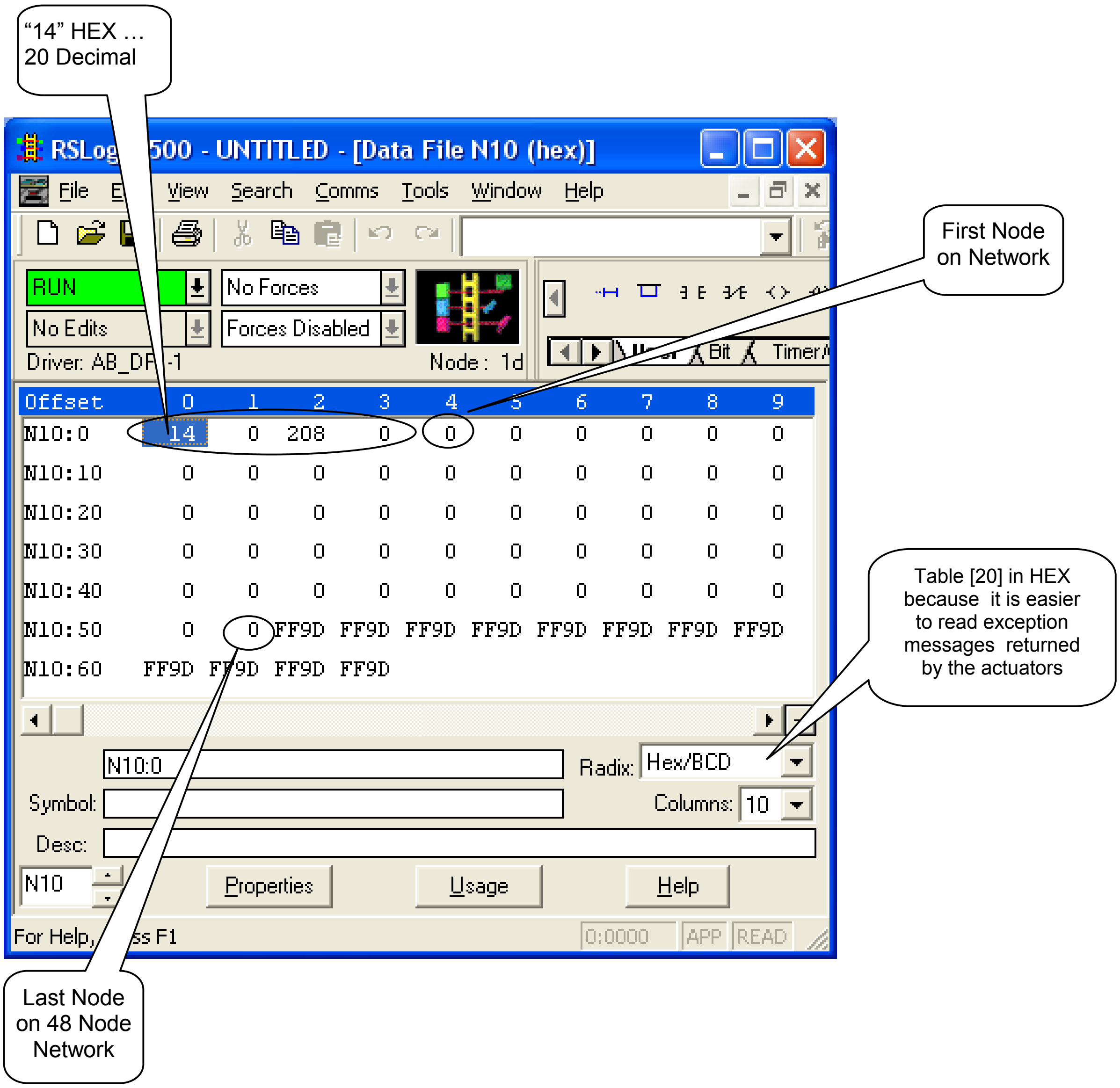
If the PLC is utilizing tables [14 or 15], it should also poll this table. There are many reasons for this. For example … if there are multiple types of EIM equipment on the network and the PLC requests data from an actuator that is "not supported by that type of actuator" (but is on the others), the actuator will return a Modbus Exception for that message and the values will NOT be loaded into tables [14 or 15]. However, the exception codes will be loaded into table [20].

Note: The intention of this table is not to identify (and maintain) this status in "real-time". It is to only serve as a general flag to the PLC/operator that an exception has occurred with this actuator and that the PLC/operator might want to investigate more thoroughly later.

Example Table [20] Feedback (48-node network) …

Characteristics:

- Table [20] is displayed hex to make it easier to interpret Modbus Exception messages returned from the actuators.
- No actuators have responded with a Modbus Exception message so far this scan period.



5.4.18. Tables [21-24]: TEC2000 Status Inputs

Words [4 → 63] PLC access: **Read / Write**

The purpose of these tables is to afford the end user all critical status information from TEC2000 systems. TEC2000 systems are backwards compatible with the 320A/B systems. However, there are more "things that can be checked" in the TEC2000. For instance, in the TEC2000, the Unit Alarm is also tripped by a "Valve Drift" Alarm. The "Valve Drift" alarm is not available in the 320A/B systems.

Therefore: *For maximum compatibility with the TEC2000 systems, it is recommended that you enable monitoring the TEC2000 Status Inputs (tables [21-24]). This way, you can ensure you have all critical status information for any TEC2000 system on the network.*

This data is derived from holding registers [1000 - 1003] in the TEC2000 actuator. These table entries are only filled in if

- The particular valve is a TEC2000 system.
- The 1746-C is configured to poll these additional registers.

This data is retrieved with a separate Modbus read command when the particular actuator is being polled (scanned).

These tables are not polled for unless the 1746-C is configured to allow it …

- if Table [0] – Word [17] == 0: The TEC2000 status input information is NOT requested when actuators are polled (*the default*).

- if Table [0] – Word [17] =/= 0: The TEC2000 status input information is requested when actuators are polled.

The PLC may reset these tables with any value is prefers. All it needs to do is insert the new value in the table and send it to the 1746-C as a write command. The table will be reset even though nothing will be sent out to any actuator.

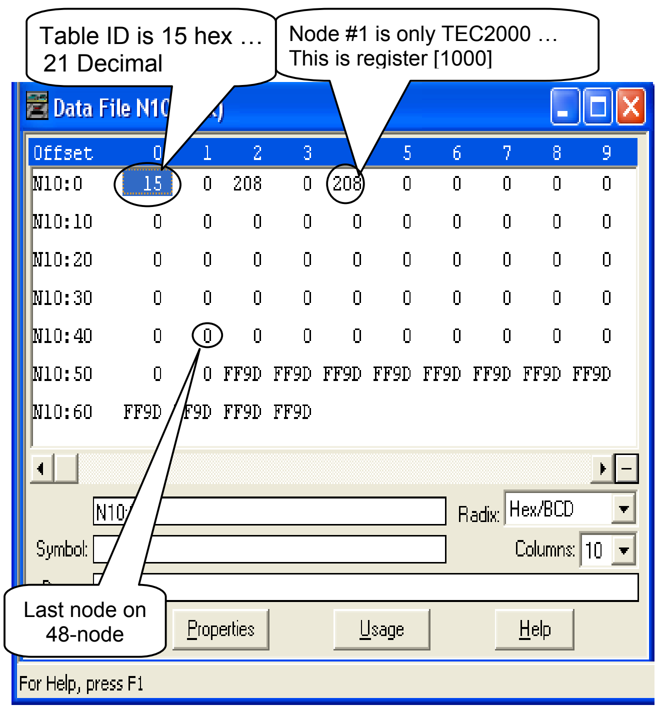
A read back of Tables [21 - 24] will

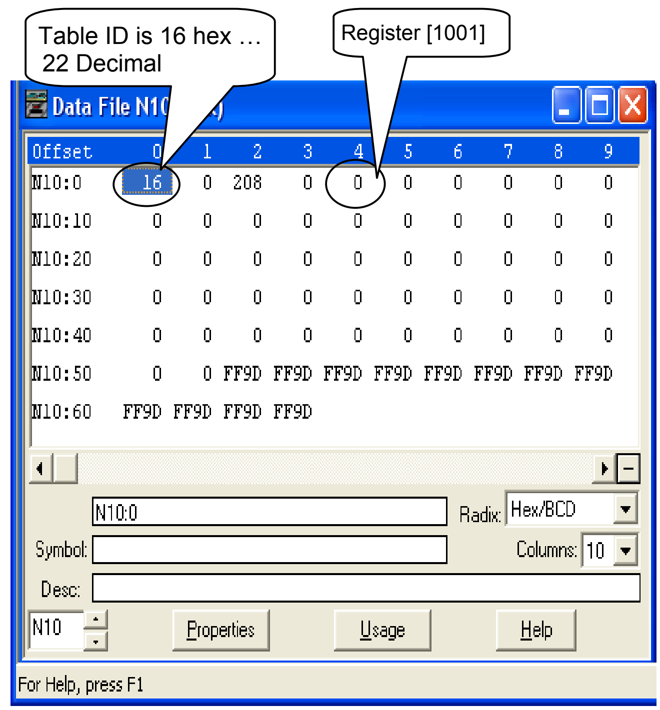
- If a **write** command issued: reflect the table write command accepted by the 1746-C
- If a **read** command issued: reflect the last known values of the status input registers in the TED2000 actuators.

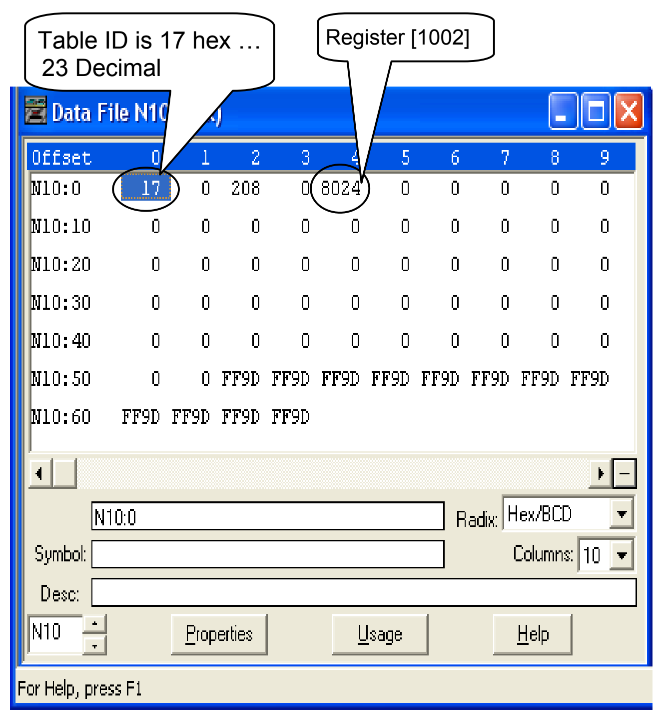
Example Tables [21-24] Feedback (48-node network) …

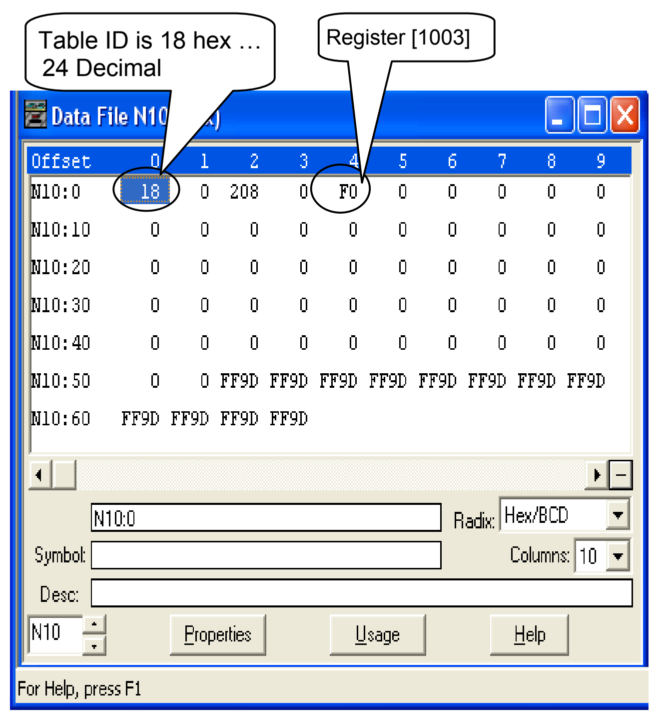
Characteristics:

- Only 1 TEC2000 system on the network … unit address #1
- Tables [21 - 24] are displayed hex because they represent bit indications … easier to read.









6. Application Notes

6.1. App Note: Performance Tuning

6.1.1. General Practices to Ensure Better Performance

To ensure optimal performance, the PLC should at least:

1. Ensure quick responses to the 1746-C requests for an M1 transfer.

   (*cuts down 1746-C delaying other operations while waiting for the PLC*)

2. Ensure table [0] is set correctly and repeated in a timely manner. Allow for frequent table [0] reads.

3. Only control & monitor discrete outputs (tables [16 & 17]) if and when needed.

   (*cuts down on the time to gather data from each actuator*)

4. Only read additional registers if and when the application program requires them.

   (*cuts down on the time to gather data from each actuator*)

5. Restrict the rate of data writes to the actuators … only write as fast as required to adequately control the valve.

   (*cuts down on the interruptions to scanning the network … gathering data*)

6. Only write to or read tables if actually in use.
   (*ex: do not write to the analog output table if analog outputs are not used*)

   (*prevents unnecessary delays to gathering the more important data from the network*)

6.1.2. Reading Run-Time Information from Table [0]

The PLC should perform a read of table [0] in a timely enough manner so that an operator can “view” the run-time changes in a manner that makes sense. For example, the operator should be able to watch the scan counter increment by 1 or 2 … if it appears to “jump” by 15 or so (because the PLC is not reading table [0] enough) then it is a lot more difficult to use the information.

6.1.3. Loading Table [0] with Configuration Information

To maximize system performance, the RLL must load configuration information that includes:

- The Number of Actuators on the network
- The Scan Period (number of network scans before a diagnostic scan).
- Whether or not to poll actuator totalizers.
- The number of additional valves to scan in the scan time-slice (before servicing the PLC).
- Table [13] … Which 320A actuators have Solid State Contactors for modulating control

- Any additional response delay for messages returning from actuators.
- Any specific register to poll from all nodes on the network.
- Any block of registers to get from a specified actuator (node) on the network.
- The value to reset the scan period counter.
- Any direct monitoring and control of discrete outputs on specific actuators on the network …
  - coils [1000 – 1011] on TEC2000 systems.
  - coils [3-5] on the other systems.
- Whether or not to poll input registers on TEC2000 systems.

The RLL is responsible for all configuration each time the system is reset. Except for 2 circumstances, all configuration is performed in table [0]. Exceptions:

1. The RLL must identify which 320A actuators require SSR control and load table [13] with the data. The optional "Solid State Relay" (SSR) is required to be in an actuator if the actuator is to receive modulating control commands from any controller. If an actuator is supposed to have an SSR installed, the network master will check, and if necessary, set the actuator internal configuration bit (indicating an SSR is installed) before sending it a modulation command.

2. The RLL must identify which actuators (if any) the network master will poll and control the discrete outputs on the actuator. This configuration information is loaded into table [16].

Upon reset, the 1746-C default configuration will assume:

- That there are 60 actuators on the network
- A Scan Period of 50 scans before a diagnostic scan.
- Do not poll the actuator totalizers.
- 2 additional valves to scan (making a total of 3 valves scanned) in the scan time-slice before servicing the PLC.
- No 320A solid-state relays contactors installed in any actuators.

- No additional response delay for messages returning from actuators.
- No specific register to poll from all nodes on the network.
- No block of registers to get from a specified actuator (node) on the network.
- The scan period reset value is zero but do not reset the counter now.
- No direct monitoring or control of discrete outputs on specific actuators on the network.
- Do not poll TEC2000 input registers.

6.1.4. Behavior if a Delay in Loading Table [0] Configuration Information

The 1746-C does not store configuration information if power is cycled. When the 1746-C restarts, it reverts to factory default values (60 actuators, 50-Scan Period, …).

It is the responsibility of the PLC to ensure that system configuration information is written to the 1746-C. The PLC should periodically read the configuration back to ensure that the 1746-C still has the correct configuration in it (the 1746-C module was not reset).

Example:

Assume a network of 25 actuators.

If power is cycled on the PLC, the 1746-C will revert to factory default values (60 actuators, …). The sooner the PLC can reload the correct Table [0] configuration, the sooner the 1746-C will start polling the correct number of units.

Until this happens, the operator will see what appears to be a “dead space” in the transmitting and receiving of data when watching the tx/rx LEDs on the 1746-C module. What happens is that:

1. A diagnostic scan has determined that only 25 of 60 actuators are on line.
2. A new scan starts.
3. 25 actuators are communicated with.
4. The 1746-C steps through the next 35 actuators … preventing communication to them because they are “off line”. This is the “dead space” when watching the LEDs.

5. The scan ends … go back to step #2 if not at the end of the scan period.

In this situation, the operator will see a very regular pattern of “dead space” on each scan (each time LED1 changes state).

Solution: Ensure the PLC writes the configuration information and periodically ensures it stays correct.

6.1.5. Using the "Scan Period" Value

The "Scan Period" value is placed in table [0] for the end user (HMI) to dynamically adjust the network behavior to the needs of the application. This value indicates the size of the "Scan Period" … the number of network scans before the 1746-C performs a diagnostic scan to try and "bring back on-line" any units that had stopped communicating.

During normal operations, the 1746-C will skip any actuator that it has already determined is "off line" (this is to save time when trying to gather data from the other actuators that are running). The only time it will bring any "off line" systems back on-line is

- During the diagnostic scan
- After reset.

If the end-user (application, HMI, ...) believes that allowing off-line units to come back on-line it is less important than normal data gathering (or just not something to be concerned with), then it should increase the scan period size (200 is the maximum ... indicating that a diagnostic scan will happen after every 200 network scans for data).

If however, the speed of units coming back on line is more critical than the increased interruption due to increasing the rate of diagnostic scans, then the HMI should lower the scan period number (5 is the minimum ... indicating that a diagnostic scan will happen after every 5 network scans for data).

In other words, recovery time to get an actuator back on line increases as the scan period value increases. Therefore, to speed up the recovery time, make the scan period value low. To lengthen the recovery time, make the scan period value high.

RECOMMENDATIONS:
Typically, EIM will not recommend any particular value for the scan period value … EIM cannot make that type of decision for any end user. However, EIM will work with a user to explain what happens if a value is chosen. The user must then make the choice of value that best suits their application. Basically, they need to evaluate the priority levels (and effects of) the time period of having an actuator off-line vs. the interruptions in the normal data gathering to perform system diagnostics (allowing off-line systems the chance of coming back on-line).

6.2. App Note: 1746-C Operation – Additional Detailed Information

6.2.1. Scan Operation: Determining & Exiting "Network Down" Condition

If the 1746-C ever detects "network down" (no actuators communicating) then LED2 will come on and the program enters the diagnostic scan mode and remains there until at least one actuator responds.

After scanning the "down" network, if any actuator responds and the number of actuators that responded is not the same as the number configured in table [0], then the 1746-C will perform one more network scan in the diagnostic mode to afford the opportunity to "simultaneously" bring as many actuators on line as possible – as soon as possible – within the same scan and not have to wait an entire scan period.

This scenario happens when "total power on a network is restored at once and the network master happened to be in the middle of a scan period when power was restored" … which is the most common scenario. In this case, the repeat diagnostic scan allows the network master the opportunity to bring "as many systems as possible on line at one time as soon as possible" and not have to wait until the end of the scan period for the next diagnostic scan.

In other words, without this feature of repeating the diagnostic scan, if the network went down and came back up in the middle of a diagnostic scan (which was most probable), then the program would usually detect only a portion of the network and would wait until the end of the scan period to check for the rest. This is a relatively long time if we are considering the network "going up/down" due network wiring and not actual powering of actuators (which was most common).
Therefore, to speed up getting most of the network on-line early, if the entire network goes down, the program repeats the diagnostic scan one more time in an attempt to get more units "on-line".

LED2 will remain on until the diagnostic scan finishes and normal scan operations resume.

6.2.2. Detailed Description of 1746-C Operation

1. Power up and self-initialization … LED1 & LED2 both on.
2. Perform the initial diagnostic scan to determine which actuators are actually on the network.
3. Turn LED1 & LED2 off.
4. Stay in a loop performing the following activities:

I. Setup to start new scan of network.

- If diagnostic scan:
  - Ensure the communication ports are setup correctly.
  - The previous scan period (group of scans) has finished.
  - Toggle the preferred communication port.
- Toggle the state of LED1 … on or off.

II. Perform the Network Scan: Using a time-slice time sharing methodology, loop through all the actuators on the network (starting from #1 and going through #n [the highest number on the network]) such that:

--- (the valve scan time-slice) -----------

a. If a diagnostic scan …
   - Clear any communication failure indications for the particular valve to be polled.
   - Get system type ID … try on both ports.
   - If an EIM system, get the firmware version ID.
b. If supposed to get totalizer data for the valve, do it.
c. If supposed to request a specific holding register, do it. If supposed to request a block of holding registers from this actuator, do it.
d. If supposed to monitor or control the discrete user relay outputs on this actuator, do it.
e. If supposed to get TEC2000 input data, do it.
f. Get the rest of the “standard data” from the valve.

   Note: If the 1746-C can’t communicate with this particular actuator that is supposed to be on-line, it marks that actuator as “off-line” … to skip on the next scan cycle. If there are problems communicating on any particular channel to the actuator, this is also recorded.

   --- (the PLC interface time-slice) -----------

g. If ready to interface with the PLC (ready for the PLC time-slice) then do it. When finished with this PLC interface time-slice, if the number of valves changed in table [0], end this loop, reset the rest of the tables and setup to enter a diagnostic scan … restart with new data.

   Otherwise still in the valve scan time-slice … If not at the last valve on the network, loop back (to step “a” above) and get data from the next actuator.

III. Network Scan cleanup

- Label Non-Used Actuator Operation. Mark all “non-used” actuator entries in the tables with an invalid value (“-99”).

- Determine if the network is “down” by counting the number of actuators that did not communicate with the 1746-C.
  If the network is down, then turn LED2 on and force the next scan to be a diagnostic scan. Otherwise ensure LED2 is off.

- Account for network scan timings to be displayed in table [0].

IV. Loop back to the top and start another network scan.

6.2.3. *Time Allocated Process Control (Allocated Time Slices)*

The "*Time Allocation Process Control*" operation acts as a process scheduler. There are two primary processes (scanning the network and servicing the PLC) and each one gets a dedicated "*time slice*". The 1746-C uses this to ensure the primary processes are serviced in a deterministic and controllable manner.

The user adjustable configuration parameter in table [0] – word [7] determines the number of "*Additional Valves to Scan in the Network Scan Time-Slice*" before starting the PLC process time slice (before interfacing with the PLC … servicing the PLC interface process).

Values for the parameter …

- The default value is 2.
  This means 2 additional valves (3 total) will be scanned in the scan time slice before the PLC process is serviced in its time slice (when a M0/M1 transfer sequence from the PLC can be handled).
- The minimum number is 0 (maximize the amount of PLC interfacing time).
- The maximum number is the lesser of
  - 9
  - 1 less than the max number of valves on the network
    (*can't scan more valves than are on the network*)

The larger the number, the faster the scan period but the fewer interfaces with the PLC.
The smaller the number, the scan period is slower but there are more frequent interfaces with the PLC.

If the number is too large, it will take too long to service the PLC and control capabilities by the HMI may fall off (ex: they could generate a "control/placement" error in which the process control can't move a modulating valve to a specified position within a specified time frame).
In other words, if too much time is allocated for scanning valves before servicing the PLC, then the PLC might not be serviced often enough for adequate control. However, servicing the PLC too often is a waste of time if it can't adequately process any differences between the services. That waste of time could be better spent polling more valves for data.

This parameter is end-user adjustable in Table [0] and is 100% dependent upon the application.
Therefore, the operator must adjust (tune) this parameter to the specific application to get the best fit of

- best scan rate
- best service rate for the PLC block transfers
- control via the PLC
- the PLC getting table information (not skipping tables)

NOTE: To help prevent accidental delays to servicing the PLC due to taking too long to scan the desired number of valves in the valve-scan time slice, a 650ms timer is in place. This timer is reset every time the 1746-C finishes servicing the PLC process. In between polling individual actuators during the valve scan time slice, if the 650ms timer ever expires then the PLC process will get serviced. In other words, there is a time-based limit to how long the network can be scanned before servicing the PLC.

Below is a table of empirical values demonstrating the difference in network performance when changing the number of additional valves to scan in the scan time slice.

This network scan scenario is for base readings with

- No Network Writes – Only Polling
- Zero M0/M1 transfer delays waiting for the RLL to interface with the 1746-C.

- 60 … Number of nodes configured to be on network
- 25 … Scan Period
- Polling base scan data … no additional registers.
- No additional delay time for Modbus response messages from the actuators.

# Nodes “Talking” … On-Line		Average Network Scan Time (seconds) after changing the number of “Additional Valves to Scan in the Network Scan Time Slice”						
		0	1	2	3	5	7	9
1 of 60 (#60)		0.2	0.2	0.2	0.2	0.2	0.2	0.2
2 of 60 (#59 – 60)		0.3	0.3	0.3	0.3	0.3	0.3	0.3
3 of 60 (#58 – 60)		0.4	0.4	0.4	0.4	0.4	0.4	0.4
6 of 60 (#55 – 60)		0.7	0.6	0.6	0.6	0.6	0.6	0.6
12 of 60 (#49 – 60)		1.3	1.2	1.1	1.1	1.0	1.0	1.0
24 of 60 (#37 – 60)		2.6	2.4	2.2	2.1	2.0	2.0	2.0
30 of 60 (#31 – 60)		3.2	3.0	2.7	2.7	2.6	2.5	2.5
36 of 60 (#25 – 60)		3.8	3.5	3.2	3.1	3.0	2.9	2.9
42 of 60 (#19 – 60)		4.4	4.1	3.8	3.7	3.6	3.5	3.4
48 of 60 (#13 – 60)		5.1	4.7	4.4	4.2	4.4	3.9	3.9
54 of 60 (#7 – 60)		5.7	5.3	4.9	4.7	4.5	4.4	4.3
60 of 60 (#1 – 60)		6.3	5.6	5.4	5.2	4.9	4.8	4.8

The values in this table “rounded values” and have a precision of (+/-) 0.2 seconds.

Because of this, these numbers are only appropriate for estimates and trend analysis.

Note: For the most accurate scan period timing, do not use the 1st scan period after a power cycle (or environment change) … wait until the 2nd scan period or beyond.

6.2.4. Preferred Communication Port Operation

The 1746-C operates on both communication ports. In a standard E>Net Ring topology, the

- Port 1 (top port) on the 1746-C goes to Port A on the EIM NIU.
- Port 2 (bottom port) on the 1746-C goes to Port B on the EIM NIU.

The implementation of this communication scheme works in conjunction with the aspect of "*Communication Failure Indications*" such that if a failure is detected on a particular port for a particular actuator, it will remain "actively announced" to assist in the diagnostics and troubleshooting required by a service technician.

The 1746-C tries to utilize both communication ports evenly. It has a "*preferred port*" and a "*backup port*" such that the program will try to communicate over the preferred port to a particular actuator. However, if there is a problem communicating with the preferred port, the program will automatically try the backup port.

The 1746-C alternates port 1 & 2 as being the preferred port after every scan period. That is, at the start of every diagnostic scan, the preferred port is changed. Example scenario:

1. Scan period n:
    - Preferred Port: Port 1
    - Backup Port: Port 2
2. Scan period n+1:
    - Preferred Port: Port 2
    - Backup Port: Port 1
3. Scan period n+2:
    - Preferred Port: Port 1
    - Backup Port: Port 2
4. ………..

If a problem is detected on the preferred port for an actuator, it will remain using the backup port until the next diagnostic scan is made to retest the port.

During restart, after the initial diagnostic scan completes, Port 2 is initially the preferred port.

6.2.5. Communication Failure Indications (Low-Level & High-Level)

Only low-level communication problems flagged as errors in tables [1 & 2].
"Modbus exception messages" are high-level (application level) communication errors. These are logged in table [20].

Both types of communication errors are cleared only at the beginning of a diagnostic scan.

The 1746-C has 3 mechanisms for reporting communication problems:

1. Table [1]: (low level) The values 0- 3 such that:
   - 0 → No problems on any port (*same as table [2] – bit-14 = 0*)
   - 1 → Problems on Port 1 (*may be used to assist diagnostics*)
   - 2 → Problems on Port 2 (*may be used to assist diagnostics*)
   - 3 → Problems on both ports … general communication fault. (*same as table [2] – bit-14 = 1*)
2. Table [2]: (low level) Bit 14 such that:
   - Bit-14 = 0: Communication OK (*same as table [1] = 0*)
   - Bit-14 = 1: General communication fault (*same as table [1] = 3*)
3. Table [20]: (high level) Modbus Exception Indicators sent from the actuators.

Notes:

- Diagnostic communication error indications for problems on port 1 or port 2 are flagged anytime a transmission is attempted and fails on a particular port.

  If an actuator is already flagged as having a problem on one port, then the program will only try using the other port (until the next diagnostic scan). If a problem also exists on the 2nd port, then a general communication fault exists.

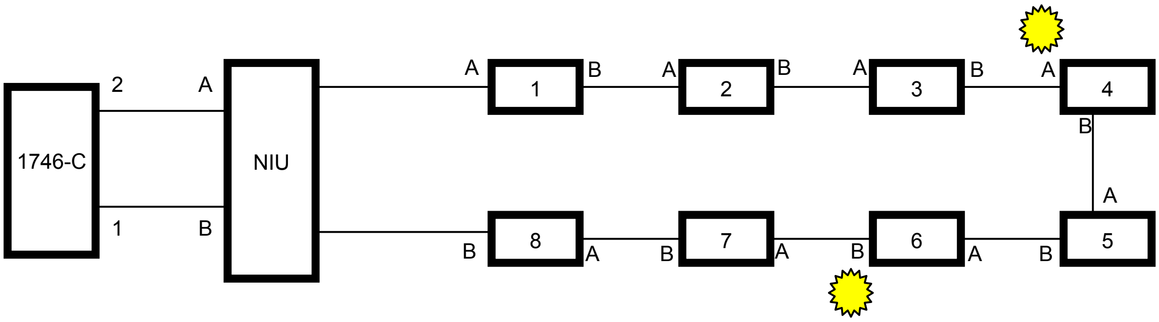
- General communication faults (Table [1] == 3) and (Table [1] – bit-14 == 1) are flagged when a message fails to successfully transmit.

- If an actuator is already flagged as having a general communication fault (can't communicate on either port) then the program will not attempt to transmit on either port (until the next diagnostic scan).

To truly understand the meaning of the values in Table [1] and the 1746-C's behavior on the network, you must know the physical layout of the network and exactly how it was wired.

For instance, at the actuators, you must know if the wiring is actually coming in from NIU-port A into port A in the actuator and out port B … (and into the next actuator in port A and out port B … … …). Actual "as built" diagrams from the electrical contractors often help here.

SAMPLE NETWORK (physical wiring):



Scenario 1: Break network at Actuator [4] – port A (between actuators 3 & 4)
In this case, the 1746-C must communicate with

- actuators 1 – 3 via port 2 and
- actuators 8 – 4 via port 1.

This means that there would be problems communicating on

- port 2 with actuators 8 – 4 and
- port 1 with actuators 1 – 3.

This means that in table [1], there would be the following values:

- “2” in the locations for actuators 8 – 4
- “1” in the locations for actuators 1 – 3.

Since all actuators could communicate, Bit-14 in table [2] would be clear for all actuators.

Scenario 2: Break network at (Actuator [4] – port A.) & (Actuator [6] – port B)
In this case, the 1746-C must communicate with

- actuators 1 – 3 via port 2 and
- actuators 8 – 7 via port 1 and
- actuators 6 – 4 cannot communicate.

This means that there would be problems communicating on

- port 2 with actuators 8 – 7 and
- port 1 with actuators 1 – 3 and
- both ports with actuators 6 – 4.

This means that in table [1], there would be the following values:

- “2” in the locations for actuators 8 – 7
- “1” in the locations for actuators 1 – 3.
- “3” in the locations for actuators 6 – 4.

Bit-14 in table [2] would be set = “1” for actuators 6 – 4. It would be clear for the rest of them.

Scenario 3: Actuator powered down.

If the actuator is just powered down and nothing else is physically wrong with the network, the network communication is supposed to pass through to the next unit. In this case, the 1746-C should be able to communicate with the entire network except for the system that is powered down.

This means that in table [1], there would be the following values:
- "3" in the locations for actuator that is powered down.
- "0" in the locations for the rest of the actuators.

Bit-14 in table [2] would be set = "1" for the actuator that is powered down. It would be clear for the rest of them.

Scenario 4: Actuator is having "problems" communicating on a particular channel (particular side … "A/B" or "Left/Right" or "In/Out" …).

In this scenario, the problems could be reflected in many ways by the 1746-C … depending upon the type of problem at the actuator. However, the highest probability is that the problem would be reflected on the appropriate 1746-C port for that actuator and any actuator that is physically located between the 1746-C and the specific actuator having problems.

For example, assume port A on actuator [4] was having a problem communicating. In this case …
Via Port 2, the 1746-C …
- can communicate with actuators 1 – 3 and
- cannot communicate with actuator 4 (by definition)
- actuators 5 – 8: depends upon the hardware problem at the actuator.

Via Port 1, the 1746-C …
- can communicate with actuators 8 – 4 and
- actuators 3 – 1: depends upon the hardware problem at the actuator.

In summary for table [1] possibilities:
- All actuators can communicate over at least one port (so no total COM faults).
  (no "3" entries in table [1])
- There is a possibility that all actuators could communicate on both channels except #4.
  (many "0" entries and possibly one "2" entry in table [1])
- There is a possibility that none of the actuators could communicate on both channels.
  (many "1" and "2" entries in table [1])

Since all actuators could communicate, Bit-14 in table [2] would be clear for all actuators.

6.2.6. Bringing Units On-Line After a Power Cycle

ACTUATOR POWER CYCLE
If you cycle power on the actuators, the 1746-C will indicate that the actuators are off line and it will only bring them back on line during the next diagnostic scan.

COMPLETE POWER CYCLE
When there is a complete power failure to the actuators and Master Station at same time (if power cycles on the entire system – actuators, PLC & 1746-C), "what happens when" depends upon what system comes back on line first.
If an actuator comes on line before the 1746-C is out of reset, then the 1746-C will see it when it performs its network check as it comes out of reset (it will be "on-line"). If the 1746-C comes up before an actuator, then that actuator will not get "on-line with the 1746-C" until the next diagnostic scan is performed.

GENERAL
Usually, if any actuators do not start communicating with the 1746-C after it performs its diagnostic scan, the places to look are outside the 1746-C (the actuators, the interconnecting network, …).

6.2.7. Toggling the Preferred Port to Assist Diagnostics

The "preferred port" toggles between PORT1 & PORT2 every time a system diagnostic scan is performed.
This timing has been shown to be long enough to give good deterministic information about communicating on any particular port and still be short enough to flag a problem that is developing on the network.
Also, since the intention behind this toggling is only to help out diagnostics (to flag an actuator when a port is starting to go bad), the relative time reference here is the time to get a technician to see and service the actuator & network.
Therefore, the scan period between diagnostic scans is more than fast enough for this (several scaling magnitudes difference ... seconds for detection as compared to minutes or hours or even days for the service technician to respond).

6.3. App Note: For Diagnostics - Know the Physical Network Wiring

When diagnosing a troublesome network, it can become impossible to solve some problems without knowing the physical layout of the network (also called “as built” by contractors). If not known, it is possible that the physical layout will have to be “traced out”.

For instance, if you are having “COM1 or COM2” problems in which you are having problems communicating in one direction and not the other (table [1] values are 1 or 2) then if you know the physical wiring of the network, it will make sense as to which ones are having problems and why. Without knowing the physical layout of the network, the information in this table can be meaningless.

Another scenario … if an E>Net ring, if the contractor accidentally swapped the Port A & B wires at one actuator, normal communications will still work fine. However, “COM1 or COM2” errors in table [1] will give incorrect indications as to where the problem lies.

6.4. App Note: Memory Maps

All references to memory maps in EIM actuators reflects the “description of the memory” as of the date of the document. To ensure correctness in the future, refer to “current” descriptions of the memory for the particular EIM product.

6.4.1. Specific Holding Registers Referenced by the Network Master

Holding Register #	Description
00	Coils [00 – 15]
01	**Not explicitly accessed by the Network Master.**
02	Coils [32 – 47]
03 – 04	**Not explicitly accessed by the Network Master.**
05	Discrete Inputs … inputs [00 – 15]
06	Valve Status Information … inputs [16 – 31]
07	Valve Position … 0 – 100% in 1% increments (Received in poll but not used)
08	Inputs [32 – 47] (Received in poll but not used)
09	Duplicate of register [06] … inputs [16 – 31] (Received in poll but not used)
10	Analog Output Register
11	Valve Postion Setpoint
12	320A (reserved) … 320B (inputs 48 – 63) (Received in poll but not used)
13	Valve Position … 0 – 100.0% in 0.1% increments
14	(AIN0) Valve Position … 0 – 4095
15	(AIN1) … Torque Pot input (Received in poll but not used)
16	(AIN2) … User Analog Input #1
17	(AIN3) … User Analog Input #2
18 - 43	**Not explicitly accessed by the Network Master.**
44	Software Version ID
45 – 65	**Not explicitly accessed by the Network Master.**
66	Totalizer (Accumulator) 1
67	Totalizer (Accumulator) 2
68 - 99	**Not explicitly accessed by the Network Master.**
100	MFG ID + Network Address
101 - 999	**Not explicitly accessed by the Network Master.**
1000	TEC2000 – Inputs [1000 – 1015]
1001	TEC2000 – Inputs [1016 – 1031]
1002	TEC2000 – Inputs [1032 – 1047]
1003	TEC2000 – Inputs [1048 – 1063]
1004	TEC2000 – Coils [1000 – 1011]

6.4.2. Specific Coils & Inputs Referenced by the Network Master

320A & 320B … (CPU reset in 320A version 1.30 & later)

Register	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0 (coils)											User Output Relay 2 RO 5	User Output Relay 1 RO 4	ESD / Monitor Relay RO 3			
2 (coils)											Enable SSR Starter R/W 37					
4 (coils)		CPU has Reset RW 78														
5 (inputs)	**Reserved 0 no-op** RO 15	User Input #2 Status RO 14	User Input #1 Status RO 13	Aux Alarm Input RO 12	Local ESD Alarm RO 11	Phase Monitor Alarm RO 10	Thermal Overload Alarm RO 9	Power Monitor Alarm RO 8	TSC Status Indicator RO 7	TSO Status Indicator RO 6	SS in Remote Position RO 5	SS in Local Position RO 4	Contactor Aux Close is Made RO 3	Contactor Aux Open is Made RO 2	LSC is Tripped RO 1	LSO is Tripped RO 0
6 (inputs)	**Unit Alarm** RO 31	0 RO 30	**Actuator Self-Test Alarm** RO 29	**Local ESD Alarm** RO 28	**Phase Monitor Alarm** RO 27	**Thermal Overload Alarm** RO 26	**Power Monitor Alarm** RO 25	**Valve Stall Alarm** RO 24	**TSC Alarm** RO 23	**TSO Alarm** RO 22	SS in Remote Position RO 21	SS in Local Position RO 20	Valve is Closing RO 19	Valve is Opening RO 18	LSC Tripped RO 17	LSO Tripped RO 16

TEC2000 …

Register	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
1000 (inputs)	**Alarm … Unit** RO 1015	STOP Command Active RO 1014	CLOSE Command Active RO 1013	Open Command Active RO 1012	SETUP Mode Selected RO 1011	REMOTE Mode Selected RO 1010	STOP Mode Selected RO 1009	LOCAL Mode Selected RO 1008	TSC is Tripped RO 1007	TSO is Tripped RO 1006	Valve is Closing RO 1005	Valve is Opening RO 1004	LSB is Tripped RO 1003	LSA is Tripped RO 1002	LSC is Tripped RO 1001	LSO is Tripped RO 1000
1001 (inputs)	**Alarm … Low Battery** RO 1031	**Alarm … Valve Stalled** RO 1030	**Alarm … Over Torque** RO 1029	Valve is Moving RO 1028	**Alarm … EFM** RO 1027	**Alarm … Config Error** RO 1026	**Alarm … Actuator Failed** RO 1025	**Alarm … Lost Analog Signal** RO 1024	**Alarm … ANY ESD** RO 1023	**Alarm … Hardwired ESD** RO 1022	**Alarm … Close Inhibit** RO 1021	**Alarm … Open Inhibit** RO 1020	**Alarm … Motor Overload** RO 1019	**Alarm … Lost Phase** RO 1018	**Alarm … Lost Ctl Voltage** RO 1017	**Alarm … Valve Drift** RO 1016
1002 (inputs)	Limits are Set Indicator RO 1047	APD Status bit RO 1046	EFM Status (bit -2 of 2) RO 1045	EFM Status (bit-1 of 2) RO 1044	ACM-Digital Net Status (bit-2 of 2) RO 1043	ACM-Digital Net Status (bit-1 of 2) RO 1042	ACM-Analog Status (bit-2 of 2) RO 1041	ACM-Analog Status (bit-1 of 2) RO 1040	AUX Close Contact Engaged RO 1039	AUX Open Contact Engaged RO 1038	Digital Input #6 State RO 1037	Digital Input #5 State RO 1036	Digital Input #4 State RO 1035	Digital Input #3 State RO 1034	Digital Input #2 State RO 1033	Digital Input #1 State RO 1032
1003 (inputs)	**Reserved NO-OP** RO 1036	**Reserved NO-OP** RO 1062	**Reserved NO-OP** RO 1061	**Reserved NO-OP** RO 1060	**Reserved NO-OP** RO 1059	**Reserved NO-OP** RO 1058	COM 2 Alarm RO 1057	COM 1 Alarm RO 1056	RDM-2 Status (bit-2 of 2) RO 1055	RDM-2 Status (bit-1 of 2) RO 1054	RDM-1 Status (bit-2 of 2) RO 1053	RDM-1 Status (bit-1 of 2) RO 1052	LDM Status (bit-2 of 2) RO 1051	LDM Status (bit-1 of 2) RO 1050	CCM Status (bit-2 of 2) RO 1049	CCM Status (bit-2 of 2) RO 1048
1004 (coils)					**RO5 CMD & RO5 Status** R/W 1011	**SO8 CMD & SO8 Status** R/W 1010	**SO7 CMD & SO7 Status** R/W 1009	**SO6 CMD & SO6 Status** R/W 1008	**RO12 CMD & Status** R/W 1007	**RO11 CMD & Status** R/W 1006	**RO10 CMD & Status** R/W 1005	**RO9 CMD & Status** R/W 1004	**RO4 CMD & Status** R/W 1003	**RO3 CMD & Status** R/W 1002	**RO2 CMD & Status** R/W 1001	**RO1 CMD & Status** R/W 1000

6.4.3. 320A Memory Map … Table for Coils & Inputs (Version 2.0)

Register	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0 (coils)	Pulsed Control Mode R/W 15	VFD Starter (Precision Ctl Mode) R/W 14	**Reserved 0 no-op** **R/W 13**	Modulating Control Mode R/W 12	**Cmd to Enter ESD Mode** **R/W 11**	**Cmd to CLOSE Valve** **R/W 10**	**Cmd to STOP Valve** **R/W 9**	**Cmd to OPEN Valve** **R/W 8**	COM 2 Enable Relay RO 7	COM 1 Enable Relay RO 6	User Output Relay 2 RO 5	User Output Relay 1 RO 4	ESD / Monitor Relay RO 3	SSR/VFD Relay RO 2	Open Valve Relay RO 1	Close Valve Relay RO 0
1 (coils)	**Unit Alarm** **RO 31**	0 RO 30	**Actuator Self-Test Alarm** **RO 29**	**Local ESD Alarm** **RO 28**	**Phase Monitor Alarm** **RO 27**	**Thermal Overload Alarm** **RO 26**	**Power Monitor Alarm** **RO 25**	**Valve Stall Alarm** **RO 24**	**TSC Alarm** **RO 23**	**TSO Alarm** **RO 22**	SS in Remote Position RO 21	SS in Local Position RO 20	Valve is Closing RO 19	Valve is Opening RO 18	LSC is tripped RO 17	LSO is tripped RO 16
2 (coils)	**Config Conflict Error** **R/W 47**	Trigger ESD on Lost Comm R/W 46	Trigger ESD on Local ESD R/W 45	Trigger ESD on Network Command R/W 44	ESD – Relay Only R/W 43	ESD – OPEN with relay R/W 42	ESD – CLOSE with relay R/W 41	ESD – OPEN No relay R/W 40	ESD – CLOSE No relay R/W 39	**Reserved 0 no-op** **R/W 38**	Enable SSR Starter R/W 37	Disable Passcode R/W 36	Enable Monitor Relay R/W 35	4-20mA position feedback on AO1 R/W 34	Enable Log Jam R/W 33	Enable Torque Seating R/W 32
3 (coils)	Relay #2 override – work with ESD R/W 63	Relay #2 NC R/W 62	Relay #1 active at LSA R/W 61	Relay #1 active at SS in Local R/W 60	Relay #1 active at SS in Remote R/W 59	Relay#1 override – work with ESD R/W 58	Relay #1 NC R/W 57	**Reserved 0 no-op** **R/W 56**	**Reserved 0 no-op** **R/W 55**	Analog Output is Calibrated R/W 54	AIN 3 is Calibrated R/W 53	AIN 2 is Calibrated R/W 52	AIN 1 is Calibrated R/W 51	Setpoint Source – AIN3 R/W 50	Setpoint Source – AIN2 R/W 49	Setpoint Source – AIN1 R/W 48
4 (coils)	**Cmd to Load Factory Defaults** **R/W 79**	CPU has Reset R/W 78	**Cmd to Save Torque Profile** **R/W 77**	Enable Back Seating R/W 76	**Reserved 0 no-op** **R/W 75**	**Reserved 0 no-op** **R/W 74**	**Reserved 0 no-op** **R/W 73**	COM 2 Alarm Enable R/W 72	COM 1 Alarm Enable R/W 71	COM 2 Even Parity R/W 70	COM 2 Odd Parity R/W 69	COM 1 Even Parity R/W 68	COM 1 Odd Parity R/W 67	Relay #2 active at LSB R/W 66	Relay #2 active at SS in Local R/W 65	Relay #2 active at SS in Remote R/W 64
5 (inputs)	**Reserved 0 no-op** **RO 15**	User Input #2 Status RO 14	User Input #1 Status RO 13	Aux Alarm Input RO 12	Local ESD Alarm RO 11	Phase Monitor Alarm RO 10	Thermal Overload Alarm RO 9	Power Monitor Alarm RO 8	TSC Status Indicator RO 7	TSO Status Indicator RO 6	SS in Remote Position RO 5	SS in Local Position RO 4	Contactor Aux Close is Made RO 3	Contactor Aux Open is Made RO 2	LSC is Tripped RO 1	LSO is Tripped RO 0
6 (inputs)	Unit Alarm RO 31	0 RO 30	Actuator Self-Test Alarm RO 29	Local ESD Alarm RO 28	Phase Monitor Alarm RO 27	Thermal Overload Alarm RO 26	Power Monitor Alarm RO 25	Valve Stall Alarm RO 24	TSC Alarm RO 23	TSO Alarm RO 22	SS in Remote Position RO 21	SS in Local Position RO 20	Valve is Closing RO 19	Valve is Opening RO 18	LSC Tripped RO 17	LSO Tripped RO 16
7	**(NOT BIT ADDRESSABLE … VALVE POSITION INDICATOR … 0 → 100% IN 1% INCREMENTS) (Read Only)**															
8 (inputs)	Phase Monitor Alarm RO 47	Local ESD Input Alarm RO 46	TSC Alarm RO 45	TSO Alarm RO 44	**COM 2 Alarm** **RO 43**	**COM 1 Alarm** **RO 42**	Power Monitor Alarm RO 41	Thermal Overload Alarm RO 40	Unit Alarm RO 39	SS in Local or Off Position RO 38	Valve Stall Alarm RO 37	Valve is Closing RO 36	Valve is Opening RO 35	Valve Stopped RO 34	LSC is Tripped RO 33	LSO is Tripped RO 32
9	START OF NON-BIT ADDRESSABLE MEMORY															

6.4.4. 320A Memory Map … Addressable Holding Registers (Version 2.0)

320A Holding Register #	Description
00	**(coils 0 → 15) … Mode / Command / Output Coils**
01	**(coils 16 → 31) … Valve status coils**
02	**(coils 32 → 47) … Configuration coils**
03	**(coils 48 → 63) … Configuration coils**
04	**(coils 64 → 79) … Configuration coils**
05	**(inputs 0 → 15) … Discrete inputs**
06	**(inputs 16 → 31) … Valve Status inputs**
07	**Valve Position**
08	**(inputs 32 → 47) … Valve Status inputs**
09	Valve status (inputs 16 → 31) … identical to register [6]
10	Analog output register (DAC)
11	Valve position setpoint
12	**RESERVED**
13	Valve position … 0 → 100% in 0.1% increments
14	AIN0 … Analog input #0 … position indicator This value indicates the actual position of the actuator.
15	AIN1 … Analog input #1 … Torque indicator pot
16	AIN2 … Analog input #2 … User analog input #1
17	AIN3 … Analog input #3 … User analog input #2
18	Water hammer setpoint
19	Modulation delay time
20	ESD delay timer
21	Position bandwidth
22	Speed bandwidth
23	**RESERVED**
24	AIN1 (Torque Pot) ZERO calibration scaling factor
25	AIN1 (Torque Pot) SPAN calibration scaling factor
26	AIN2 (User AIN #1) ZERO calibration scaling factor
27	AIN2 (User AIN #1) SPAN calibration scaling factor
28	AIN3 (User AIN #2) ZERO calibration scaling factor
29	AIN3 (User AIN #2) SPAN calibration scaling factor
30	Analog output ZERO calibration scaling factor
31	Analog output SPAN calibration scaling factor
32	LSA setpoint
33	LSB setpoint
34	Valve CLOSE Duty Cycle Timer – ON Time
35	Valve CLOSE Duty Cycle Timer – OFF Time
36	Valve OPEN Duty Cycle Timer – ON Time
37	Valve OPEN Duty Cycle Timer – OFF Time
38	COM1 baud rate
39	COM1 response delay
40	COM2 baud rate
41	COM2 response delay
42	Passcode characters 1&2 … (1 → LSB) (2 → MSB)
43	Passcode characters 3&4 … (3 → LSB) (4 → MSB)
44	Software Version ID
45	**RESERVED**

320A Holding Register #	Description
46	**RESERVED**
47	**RESERVED**
48	Current torque at 10% open while opening valve
49	Current torque at 20% open while opening valve
50	Current torque at 30% open while opening valve
51	Current torque at 40% open while opening valve
52	Current torque at 50% open while opening valve
53	Current torque at 60% open while opening valve
54	Current torque at 70% open while opening valve
55	Current torque at 80% open while opening valve
56	Current torque at 90% open while opening valve
57	Archived torque at 10% open while opening valve … EEPROM
58	Archived torque at 20% open while opening valve … EEPROM
59	Archived torque at 30% open while opening valve … EEPROM
60	Archived torque at 44% open while opening valve … EEPROM
61	Archived torque at 50% open while opening valve … EEPROM
62	Archived torque at 60% open while opening valve … EEPROM
63	Archived torque at 70% open while opening valve … EEPROM
64	Archived torque at 80% open while opening valve … EEPROM
65	Archived torque at 90% open while opening valve … EEPROM
66	User Digital Input #1 Totalizer (accumulator)
67	User Digital Input #2 Totalizer (accumulator)
68	Loss-of-COM alarm delay time.
69	Valve Stall delay time
70	**RESERVED**
71 – 99	**Not used … returns an error if accessed.**
100	MFG ID + Network Address

6.4.5. 320B Memory Map … Table for Coils & Inputs

Register	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0 (coils)	Pulsed Control Mode R/W 15	VFD Starter (Precision Ctl Mode) R/W 14	**Reserved 0 no-op** R/W 13	Modulating Control indicator (CAM08) R/W 12	**Cmd to Enter ESD Mode** R/W 11	**Cmd to CLOSE Valve** R/W 10	**Cmd to STOP Valve** R/W 9	**Cmd to OPEN Valve** R/W 8	COM 2 Enable Relay RO 7	COM 1 Enable Relay RO 6	User Output Relay 2 RO 5	User Output Relay 1 RO 4	ESD / Monitor Relay RO 3	SSR/VFD Relay RO 2	Open Valve Relay RO 1	Close Valve Relay RO 0
1 (coils)	Unit Alarm RO 31	0 RO 30	Actuator Self-Test Alarm RO 29	Local ESD Alarm RO 28	Phase Monitor Alarm RO 27	Thermal Overload Alarm RO 26	Power Monitor Alarm RO 25	Valve Stall Alarm RO 24	TSC Alarm RO 23	TSO Alarm RO 22	SS in Remote Position RO 21	SS in Local Position RO 20	Valve is Closing RO 19	Valve is Opening RO 18	LSC is tripped RO 17	LSO is tripped RO 16
2 (coils)	**Config Conflict Error** R/W 47	Trigger ESD on Lost Comm R/W 46	Trigger ESD on Local ESD R/W 45	Trigger ESD on Network Command R/W 44	ESD – STOP with Relay R/W 43	ESD – OPEN with Relay R/W 42	ESD – CLOSE with Relay R/W 41	ESD – OPEN No relay R/W 40	ESD – CLOSE No relay R/W 39	**Reserved 0 no-op** R/W 38	Enable SSR Starter R/W 37	Passcode Enabled R/W 36	Enable Monitor Relay Operations R/W 35	4-20mA position feedback on AO1 R/W 34	Enable Log Jam R/W 33	Enable Torque Seating R/W 32
3 (coils)	Relay #2 override – work with ESD R/W 63	Relay #2 NC R/W 62	Relay #1 active at LSA R/W 61	Relay #1 active at SS in Local R/W 60	Relay #1 active at SS in Remote R/W 59	Relay#1 override – work with ESD R/W 58	Relay #1 NC R/W 57	**Reserved 0 no-op** R/W 56	**Reserved 0 no-op** R/W 55	**Reserved 0 no-op** R/W 54	AIN 3 Fault Move to Default R/W 53	AIN 2 Fault Move to Default R/W 52	AIN 1 Fault Move to Default R/W 51	Setpoint Source – AIN3 R/W 50	Setpoint Source – AIN2 R/W 49	Setpoint Source – AIN1 R/W 48
4 (coils)	**Cmd - Load Factory Defaults** R/W 79	CPU has Reset R/W 78	**Cmd - Save Torque Profile** R/W 77	Enable Back Seating R/W 76	Enable MRTU Operations R/W 75	**Reserved 0 no-op** R/W 74	**Reserved 0 no-op** R/W 73	COM 2 Alarm Enable R/W 72	COM 1 Alarm Enable R/W 71	COM 2 Even Parity R/W 70	COM 2 Odd Parity R/W 69	COM 1 Even Parity R/W 68	COM 1 Odd Parity R/W 67	Relay #2 active at LSB R/W 66	Relay #2 active at SS in Local R/W 65	Relay #2 active at SS in Remote R/W 64
5 (inputs)	**Reserved 0 no-op** RO 15	User Input #2 Status RO 14	User Input #1 Status RO 13	Aux Alarm Input RO 12	Local ESD Alarm RO 11	Phase Monitor Alarm RO 10	Thermal Overload Alarm RO 9	Power Monitor Alarm RO 8	TSC Status Indicator RO 7	TSO Status Indicator RO 6	SS in Remote Position RO 5	SS in Local Position RO 4	Contactor Aux Close is Made RO 3	Contactor Aux Open is Made RO 2	LSC is Tripped RO 1	LSO is Tripped RO 0
6 (inputs)	**Unit Alarm** RO 31	0 RO 30	**Actuator Self-Test Alarm** RO 29	**Local ESD Alarm** RO 28	**Phase Monitor Alarm** RO 27	**Thermal Overload Alarm** RO 26	**Power Monitor Alarm** RO 25	**Valve Stall Alarm** RO 24	**TSC Alarm** RO 23	**TSO Alarm** RO 22	SS in Remote Position RO 21	SS in Local Position RO 20	Valve is Closing RO 19	Valve is Opening RO 18	LSC Tripped RO 17	LSO Tripped RO 16
7	**(NOT BIT ADDRESSABLE … VALVE POSITION INDICATOR … 0 → 100% IN 1% INCREMENTS) (Read Only)**															
8 (inputs)	Phase Monitor Alarm RO 47	Local ESD Input Alarm RO 46	TSC Alarm RO 45	TSO Alarm RO 44	**COM 2 Alarm** RO 43	**COM 1 Alarm** RO 42	Power Monitor Alarm RO 41	Thermal Overload Alarm RO 40	Unit Alarm RO 39	SS in Local or Off Position RO 38	Valve Stall Alarm RO 37	Valve is Closing RO 36	Valve is Opening RO 35	Valve Stopped RO 34	LSC is Tripped RO 33	LSO is Tripped RO 32
9 – 11	NON-BIT ADDRESSABLE MEMORY															
12 (inputs)	**Reserved no-op** RO 63	**Reserved no-op** RO 62	**Reserved no-op** RO 61	**Reserved no-op** RO 60	**Reserved no-op** RO 59	**Reserved no-op** RO 58	**Reserved no-op** RO 57	**Reserved no-op** RO 56	**Reserved no-op** RO 55	**Reserved no-op** RO 54	**Reserved no-op** RO 53	**Software Based ESD is Active** RO 52	**AIN 3 Signal Fault** RO 51	**AIN 2 Signal Fault** RO 50	**AIN1 Signal Fault** RO 49	SETUP Mode Indicator RO 48
13	END OF BIT ADDRESSABLE MEMORY															

6.4.6. 320B Memory Map … Addressable Holding Registers

320B Holding Register #	Description	Range (Remote Interface)	Increment (Remote Interface)
00	**(coils 0 → 15) … Mode / Command / Output Coils**	**-----------**	**----------**
01	**(coils 16 → 31) … Valve status coils**	**-----------**	**----------**
02	**(coils 32 → 47) … Configuration coils**	**-----------**	**----------**
03	**(coils 48 → 63) … Configuration coils**	**-----------**	**----------**
04	**(coils 64 → 79) … Configuration coils**	**-----------**	**----------**
05	**(inputs 0 → 15) … Discrete inputs**	**-----------**	**----------**
06	**(inputs 16 → 31) … Valve Status inputs**	**-----------**	**----------**
07	**Valve Position**	0 – 100	1%
08	**(inputs 32 → 47) … Valve Status inputs**	**-----------**	**----------**
09	Valve status (inputs 16 → 31) … identical to register [6]	-----------	----------
10	Analog output register (DAC)	0 – 4095	Analog counts
11	Valve position setpoint	0 – 4095	Analog counts
12	**(Inputs 48 → 63)**	**-----------**	**----------**
13	Valve position … 0 → 100% in 0.1% increments	0 – 1000	0.1%
14	AIN0 … Analog input #0 … position indicator. This value indicates the actual position of the actuator.	0 – 4095	Analog counts
15	AIN1 … Analog input #1 … Torque indicator pot	0 – 4095	Analog counts
16	AIN2 … Analog input #2 … User analog input #1	0 – 4095	Analog counts
17	AIN3 … Analog input #3 … User analog input #2	0 – 4095	Analog counts
18	Water hammer setpoint	0 – 4095	Analog counts
19	Modulation delay time	100 – 25500	0.001 second
20	ESD delay timer	0 – 65535	0.001 second
21	Position bandwidth	4 – 205	Analog counts
22	Speed bandwidth	12 – 410	Analog counts
23	Default Setpoint on Analog Input Fault Indication	0 – 4095	Analog counts
24	AIN1 (Torque Pot) ZERO calibration scaling factor	0 – 4095	Analog counts
25	AIN1 (Torque Pot) SPAN calibration scaling factor	0 – 4095	Analog counts
26	AIN2 (User AIN #1) ZERO calibration scaling factor	0 – 4095	Analog counts
27	AIN2 (User AIN #1) SPAN calibration scaling factor	0 – 4095	Analog counts
28	AIN3 (User AIN #2) ZERO calibration scaling factor	0 – 4095	Analog counts
29	AIN3 (User AIN #2) SPAN calibration scaling factor	0 – 4095	Analog counts
30	Analog output ZERO calibration scaling factor	0 – 4095	Analog counts
31	Analog output SPAN calibration scaling factor	0 – 4095	Analog counts
32	LSA setpoint	0 – 4095	Analog counts
33	LSB setpoint	0 – 4095	Analog counts
34	Valve CLOSE Duty Cycle Timer – ON Time	0 – 65535	0.001 second
35	Valve CLOSE Duty Cycle Timer – OFF Time	0 – 65535	0.001 second
36	Valve OPEN Duty Cycle Timer – ON Time	0 – 65535	0.001 second
37	Valve OPEN Duty Cycle Timer – OFF Time	0 – 65535	0.001 second
38	COM1 baud rate	1200 – 38400	Baud rate
39	COM1 response delay	8 – 60	0.001 second
40	COM2 baud rate	1200 – 38400	Baud rate
41	COM2 response delay	8 – 60	0.001 second
42	Passcode characters 1&2 … (1 → LSB) (2 → MSB)	ASCII	ASCII
43	Passcode characters 3&4 … (3 → LSB) (4 → MSB)	ASCII	ASCII
44	Software Version ID	---	---

320B Holding Register #	Description	Range (Remote Interface)	Increment (Remote Interface)
45	Open coast distance	0 – 4095	Analog counts
46	Close coast distance	0 – 4095	Analog counts
47	**Reserved**	**-----------**	**----------**
48	Current torque at 10% open while opening valve	0 – 4095	Analog counts
49	Current torque at 20% open while opening valve	0 – 4095	Analog counts
50	Current torque at 30% open while opening valve	0 – 4095	Analog counts
51	Current torque at 40% open while opening valve	0 – 4095	Analog counts
52	Current torque at 50% open while opening valve	0 – 4095	Analog counts
53	Current torque at 60% open while opening valve	0 – 4095	Analog counts
54	Current torque at 70% open while opening valve	0 – 4095	Analog counts
55	Current torque at 80% open while opening valve	0 – 4095	Analog counts
56	Current torque at 90% open while opening valve	0 – 4095	Analog counts
57	Archived torque at 10% open while opening valve … EEPROM	0 – 4095	Analog counts
58	Archived torque at 20% open while opening valve … EEPROM	0 – 4095	Analog counts
59	Archived torque at 30% open while opening valve … EEPROM	0 – 4095	Analog counts
60	Archived torque at 44% open while opening valve … EEPROM	0 – 4095	Analog counts
61	Archived torque at 50% open while opening valve … EEPROM	0 – 4095	Analog counts
62	Archived torque at 60% open while opening valve … EEPROM	0 – 4095	Analog counts
63	Archived torque at 70% open while opening valve … EEPROM	0 – 4095	Analog counts
64	Archived torque at 80% open while opening valve … EEPROM	0 – 4095	Analog counts
65	Archived torque at 90% open while opening valve … EEPROM	0 – 4095	Analog counts
66	User Digital Input #1 Totalizer (accumulator)	0 – 65535	Input strobes
67	User Digital Input #2 Totalizer (accumulator)	0 – 65535	Input strobes
68	Loss-of-COM alarm delay time.	1000 – 65535	0.001 second
69	Valve Stall delay time	400 – 65535	0.001 second
70	**Travel time for 1% travel**	100 – 36000	0.001 second
71	**Reserved**	**---**	**---**
72 – 99	**Not used … returns an error if accessed.**	**---**	**---**
100	MFG ID + Network Address	---	---

6.4.7. TEC2000 Memory Map … Table for Coils & Inputs

Register	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
1000 (inputs)	Alarm … Unit RO 1015	STOP Command Active RO 1014	CLOSE Command Active RO 1013	Open Command Active RO 1012	SETUP Mode Selected RO 1011	REMOTE Mode Selected RO 1010	STOP Mode Selected RO 1009	LOCAL Mode Selected RO 1008	Alarm … TSC (Close) RO 1015	Alarm … TSO (Open) RO 1014	Valve is Closing RO 1013	Valve is Opening RO 1012	LSB is Tripped RO 1011	LSA is Tripped RO 1010	LSC is Tripped RO 1009	LSO is Tripped RO 1008
1001 (inputs)	Alarm … Low Battery RO 1031	Alarm … Valve Stalled RO 1030	Alarm … Over Torque RO 1029	Valve is Moving RO 1028	COM 2 Alarm RO 1027	COM 1 Alarm RO 1026	Alarm … System Service Required RO 1025	Alarm … Lost Control Signal RO 1024	Alarm … Any Non-Hardwired ESD RO 1031	Alarm … Hardwired ESD RO 1030	Alarm … Close Inhibit RO 1029	Alarm … Open Inhibit RO 1028	Alarm … Motor Overload RO 1027	Alarm … Lost Phase RO 1026	Alarm … Lost Control Voltage Signal RO 1025	Alarm … Valve Drift RO 1024
1002 (inputs)	Sleep Mode Active RO 1047	Emergency STOP Signal Status RO 1046	100% CW Toque Signal Status RO 1045	100% CCW Toque Signal Status RO 1044	Operating Inhibit Signal Status RO 1043	Torque Inhibit Signal Status RO 1042	AIN2 Signal Fault RO 1041	AIN1 Signal Fault RO 1040	AUX Close Contact Engaged RO 1047	AUX Open Contact Engaged RO 1046	Digital Input #6 State RO 1045	Digital Input #5 State RO 1044	Digital Input #4 State RO 1043	Digital Input #3 State RO 1042	Digital Input #2 State RO 1041	Digital Input #1 State RO 1040
1003 (inputs)	Limits are NOT Set Indicator RO 1063	Alarm … CCM Link Lost to APD RO 1062	Alarm … APD Failed RO 1061	Alarm … Actuator Rotation Fault RO 1060	Alarm … CCM Link Lost to ACM RO 1059	Alarm … ACM Failed RO 1058	Alarm … EFM Signal Status RO 1057	Alarm … Config Error RO 1056	Alarm … CCM Link Lost to RDM2 RO 1063	Alarm … RDM2 Failed RO 1062	Alarm … CCM Link Lost to RDM1 RO 1061	Alarm … RDM1 Failed RO 1060	Alarm … CCM Link Lost to LDM RO 1059	Alarm … LDM Failed RO 1058	CCM Status (bit-2 of 2) RO 1057	CCM Status (bit-1 of 2) RO 1056
1004 (coils)	Rmt Cmd - Enter ESD R/W 1015	CCM CPU Has reset R/W 1014	Reserved NO-OP R/W 1013	Reserved NO-OP R/W 1012	RO5 CMD & RO5 Status R/W 1011	SO8 CMD & SO8 Status R/W 1010	SO7 CMD & SO7 Status R/W 1009	SO6 CMD & SO6 Status R/W 1008	RO12 CMD & Status R/W 1015	RO11 CMD & Status R/W 1014	RO10 CMD & Status R/W 1013	RO9 CMD & Status R/W 1012	RO4 CMD & Status R/W 1011	RO3 CMD & Status R/W 1010	RO2 CMD & Status R/W 1009	RO1 CMD & Status R/W 1008
1005 (coils)	CMD – Load Field Setup Defaults R/W 1031	CMD – STOP Valve R/W 1030	CMD – CLOSE Valve R/W 1029	CMD – OPEN Valve R/W 1028	CMD - enter SETUP mode R/W 1027	COM2 Fault Indicator R/W 1026	COM1 Fault Indicator R/W 1025	AIN2 Fault Indicator R/W 1024	AIN1 Fault Indicator R/W 1031	Reserved NO-OP R/W 1030	Reserved NO-OP R/W 1029	Reserved NO-OP R/W 1028	Reserved NO-OP R/W 1027	Reserved NO-OP R/W 1026	Reserved NO-OP R/W 1025	Reserved NO-OP R/W 1024
1006 (coils)	CMD – Clear Alarm Log R/W 1047	Reserved - No write Access RO 1046	CMD – Archive Torque Profile R/W 1045	Reserved - No write Access RO 1044	Reserved - No write Access RO 1043	Reserved - No write Access RO 1042	Reserved - No write Access RO 1041	Reserved - No write Access RO 1040	CMD – Clear Limits R/W 1047	CMD – Save Travel Limits RO 1046	CMD – Accept Port C Limit R/W 1045	CMD – Accept OPEN Limit Port B Limit RO 1044	CMD – Accept Close Limit Port A Limit RO 1043	CMD – Clear All Archives RO 1042	CMD – Clear All Logs & Profile RO 1041	CMD – Load Factory Defaults RO 1040
1007 (coils)	Reserved NO-OP R/W 1063	Reserved NO-OP R/W 1062	Reserved NO-OP R/W 1061	Reserved NO-OP R/W 1060	Reserved NO-OP R/W 1059	Alternate Language Selected R/W 1058	COM 2 Alarm Enable R/W 1057	COM1 Alarm Enable R/W 1056	Reserved NO-OP R/W 1063	APD Type R/W 1062	Line (Motor) Frequency (bit 2 of 2) R/W 1061	Line (Motor) Frequency (bit 1 of 2) R/W 1060	3-Phase R/W 1059	Starter Type (bit 2 of 2) R/W 1058	Starter Type (bit 1 of 2) R/W 1057	Multi-Turn Valve Installation R/W 1056
1008 (coils)	Setpoint Source (if AIN control) R/W 1079	Enable Open Torque Back Seating R/W 1078	Enable Close Torque Seating R/W 1077	Enable Local Seal-In R/W 1076	Enable Remote Seal-In R/W 1075	Enable Torque Retry R/W 1074	RDM1 Installed R/W 1073	RDM2 Installed R/W 1072	ESD Override – LOCAL Mode Ctl R/W 1079	ESD Override – STOP Mode Ctl R/W 1078	ESD Override – Inhibits R/W 1077	ESD Override – Torque Alarm R/W 1076	ARM is Installed R/W 1075	Enable Open Cycle Timer R/W 1074	Enable Close Cycle Timer R/W 1073	Enable Anti-Water Hammer R/W 1072
1009 (coils)	Controlinc ACM Installed R/W 1095	Futronic ACM Installed R/W 1094	AO2 Source (bit 1 of 2) R/W 1093	AO1 Source (bit 2 of 2) R/W 1092	AO1 Source (bit 1 of 2) R/W 1091	Multi-Port Valve Installation R/W 1090	Drive Sleeve Closes in CW direction R/W 1089	Enable Multi-Port Synch R/W 1088	Enable Inhibit Open Valve Operations R/W 1095	Enable Inhibit Close Valve Operations R/W 1094	ESD Trigger on Hardwired (LOCAL) ESD R/W 1093	ESD Trigger on Remote Host Command R/W 1092	ESD Trigger on Loss of Control R/W 1091	ESD Action (bit 2 of 2) R/W 1090	ESD Action (bit 1 of 2) R/W 1089	ESD Override – Thermal Overload Alarm R/W 1088
1010 (coils)	Worm Gear is LH R/W 1111	Set LED … Open – RD Close – GR R/W 1110	Reserved NO-OP R/W 1109	Motor Rotation is Electrically Reversed R/W 1108	Enable Battery Alarm R/W 1107	AO2 Source (bit 2 of 2) R/W 1106	Enable AIN2 Alarm on Lost Signal R/W 1105	Enable AIN1 Alarm on Lost Signal R/W 1104	Action if Lost Control Signal (bit 2 of 2)l R/W 1111	Action if Lost Control Signal (bit 1 of 2)l R/W 1110	Reserved NO-OP R/W 1109	Analog Output "Close" Polarity R/W 1108	Analog Input "Close" Polarity R/W 1107	Control Mode (bit 3 of 3) R/W 1106	Control Mode (bit 2 of 3) R/W 1105	Control Mode (bit 1 of 3) R/W 1104

6.4.8. TEC2000 Memory Map … Table for Holding Registers

Description		
Reg[1000] ... Discrete Inputs 1000 – 1015		
Reg[1001] ... Discrete Inputs 1016 – 1031		
Reg[1002] ... Discrete Inputs 1032 – 1047		
Reg[1003] ... Discrete Inputs 1048 – 1063		
Reg[1004] ... Coils 1000-1015		
Reg[1005] ... Coils 1016-1031		
Reg[1006] ... Coils 1032-1047		
Reg[1007] ... Coils 1048-1063		
Reg[1008] ... Coils 1064-1079		
Reg[1009] ... Coils 1080-1095		
Reg[1010] ... Coils 1096-1111		
GENERAL DESCRIPTION	**HIGH BYTE**	**LOW BYTE**
Reg[1011] – Factory Setup	Model ID - Char 1	Model ID - Char 0
Reg[1012] – Factory Setup	Model ID - Char 3	Model ID - Char 3
Reg[1013] – Factory Setup	Model ID - Char 5	Model ID - Char 4
Reg[1014] – Factory Setup	Serial Num - Char 0	Model ID - Char 6
Reg[1015] – Factory Setup	Serial Num - Char 2	Serial Num – Char 1
Reg[1016] – Factory Setup	Serial Num - Char 4	Serial Num – Char 3
Reg[1017] – Factory Setup	Serial Num - Char 6	Serial Num – Char 5
Reg[1018] – Factory Setup	Serial Num - Char 8	Serial Num – Char 7
Reg[1019] – Factory Setup	MFG Date - Char 0	Serial Num – Char 9
Reg[1020] – Factory Setup	MFG Date - Char 2	MFG Date - Char 1
Reg[1021] – Factory Setup	Motor ID - Char 0	MFG Date - Char 3
Reg[1022] – Factory Setup	Motor ID - Char 2	Motor ID - Char 1
Reg[1023] – Factory Setup	Motor ID - Char 4	Motor ID - Char 3
Reg[1024] – Factory Setup	Motor ID - Char 6	Motor ID - Char 5
Reg[1025] – Factory Setup	Motor ID - Char 8	Motor ID - Char 7
Reg[1026] – Factory Setup	Motor HP - Char 0	Motor ID - Char 9
Reg[1027] – Factory Setup	Motor HP - Char 2	Motor HP - Char 1
Reg[1028] – Factory Setup	Motor HP - Char 4	Motor HP - Char 3
Reg[1029] – Factory Setup	Motor RPM - Char 1	Motor RPM - Char 0
Reg[1030] – Factory Setup	Motor RPM - Char 3	Motor RPM - Char 2
Reg[1031] – Factory Setup	Motor Run AMPS - Char 1	Motor Run AMPS - Char 0
Reg[1032] – Factory Setup	Motor Run AMPS - Char 3	Motor Run AMPS - Char 2
Reg[1033] – Factory Setup	Motor Stall AMPS - Char 1	Motor Stall AMPS - Char 0
Reg[1034] – Factory Setup	Motor Stall AMPS - Char 3	Motor Stall AMPS - Char 2
Reg[1035] – Factory Setup	Max Torque (0 – 100%)	Motor Voltage
Reg[1036] – Factory Setup	Torque Spring Type	CAM Card Type (Network)
Reg[1037] – User (Field) Setup	RO2 Function	RO1 Function
Reg[1038] – User (Field) Setup	RO4 Function	RO3 Function
Reg[1039] – User (Field) Setup	RO9 Function	RO5 Function
Reg[1040] – User (Field) Setup	RO11 Function	RO10 Function
Reg[1041] – User (Field) Setup	Field Setup Pass Code – Char 0	RO12 Function
Reg[1042] – User (Field) Setup	Field Setup Pass Code – Char 2	Field Setup Pass Code – Char 1
Reg[1043] – User (Field) Setup	Max Close Torque (0 – 100%)	Max Open Torque (0 – 100%)
Reg[1044] – User (Field) Setup	Tag ID – Char 1	Tag ID – Char 0
Reg[1045] – User (Field) Setup	Tag ID – Char 3	Tag ID – Char 2
Reg[1046] – User (Field) Setup	Tag ID – Char 5	Tag ID – Char 4

Reg[1047] – User (Field) Setup	Tag ID – Char 7	Tag ID – Char 6
Reg[1048] – User (Field) Setup	Tag ID – Char 9	Tag ID – Char 8
Reg[1049] – User (Field) Setup	Tag ID – Char 11	Tag ID – Char 10
Reg[1050] – User (Field) Setup	Tag ID – Char 13	Tag ID – Char 12
Reg[1051] – User (Field) Setup	Tag ID – Char 15	Tag ID – Char 14
Reg[1052] – User (Field) Setup	LSB Setting	LSA Setting
Reg[1053] – User (Field) Setup	Close Timer Start Position	Close Timer Stop Position
Reg[1054] – User (Field) Setup	Close Timer Pulse ON Time	Close Timer Pulse OFF Time
Reg[1055] – User (Field) Setup	Open Timer Start Position	Open Timer Stop Position
Reg[1056] – User (Field) Setup	Open Timer Pulse ON Time	Open Timer Pulse OFF Time
Reg[1057] – User (Field) Setup	Water Hammer Start Position	Water Hammer Pulse ON Time
Reg[1058] – User (Field) Setup	Water Hammer Pulse Off Time	Default Setpoint Position
Reg[1059] – User (Field) Setup	Network Node Address	Network Response Delay
Reg[1060] – User (Field) Setup	Speed Control Bandwidth	Position Bandwidth
Reg[1061] – User (Field) Setup	ESD Delay Time	Modulation Delay Time
Reg[1062] – User (Field) Setup	Event Log Index	Discrete Input Inversion Bits
Reg[1063] – User (Field) Setup	**(RESERVED – NO-OP)**	Loss-of-COM Alarm Delay Time
Reg[1064] – User (Field) Setup	Date/Time Clock – Minutes	Date/Time Clock – Seconds
Reg[1065] – User (Field) Setup	Date/Time Clock – Day	Date/Time Clock – Hour
Reg[1066] – User (Field) Setup	Date/Time Clock – Year	Date/Time Clock – Month
Reg[1067] – Real-Time Data	Current Torque (0-100%)	Current Position (0-100%)
Reg[1068] – Real-Time Data	Last Open Torque at (3-10%)	Last Open Torque at (0-2%)
Reg[1069] – Real-Time Data	Last Open Torque at (21-30%)	Last Open Torque at (11-20%)
Reg[1070] – Real-Time Data	Last Open Torque at (41-50%)	Last Open Torque at (31-40%)
Reg[1071] – Real-Time Data	Last Open Torque at (61-70%)	Last Open Torque at (51-60%)
Reg[1072] – Real-Time Data	Last Open Torque at (81-90%)	Last Open Torque at (71-80%)
Reg[1073] – Real-Time Data	Last Open Torque at (98-100%)	Last Open Torque at (91-97%)
Reg[1074] – Real-Time Data	Last Close Torque at (3-10%)	Last Close Torque at (0-2%)
Reg[1075] – Real-Time Data	Last Close Torque at (21-30%)	Last Close Torque at (11-20%)
Reg[1076] – Real-Time Data	Last Close Torque at (41-50%)	Last Close Torque at (31-40%)
Reg[1077] – Real-Time Data	Last Close Torque at (61-70%)	Last Close Torque at (51-60%)
Reg[1078] – Real-Time Data	Last Close Torque at (81-90%)	Last Close Torque at (71-80%)
Reg[1079] – Real-Time Data	Last Close Torque at (98-100%)	Last Close Torque at (91-97%)
Reg[1080] – Archived Torque	Open Torque at (3-10%)	Open Torque at (0-2%)
Reg[1081] – Archived Torque	Open Torque at (21-30%)	Open Torque at (11-20%)
Reg[1082] – Archived Torque	Open Torque at (41-50%)	Open Torque at (31-40%)
Reg[1083] – Archived Torque	Open Torque at (61-70%)	Open Torque at (51-60%)
Reg[1084] – Archived Torque	Open Torque at (81-90%)	Open Torque at (71-80%)
Reg[1085] – Archived Torque	Open Torque at (98-100%)	Open Torque at (91-97%)
Reg[1086] – Archived Torque	Close Torque at (3-10%)	Close Torque at (0-2%)
Reg[1087] – Archived Torque	Close Torque at (21-30%)	Close Torque at (11-20%)
Reg[1088] – Archived Torque	Close Torque at (41-50%)	Close Torque at (31-40%)
Reg[1089] – Archived Torque	Close Torque at (61-70%)	Close Torque at (51-60%)
Reg[1090] – Archived Torque	Close Torque at (81-90%)	Close Torque at (71-80%)
Reg[1091] – Archived Torque	Close Torque at (98-100%)	Close Torque at (91-97%)
Reg[1092] – Log	Motor Run-time Log – Min	Motor Run-Time Log – Sec
Reg[1093] – Log	**(RESERVED – NO-OP)**	Motor Run-Time Log – Hours
Reg[1094] – Log	Motor Starts Log (Word 1)	
Reg[1095] – Log	Motor Starts Log (Word 2)	
Reg[1096] – Log	Motor Starts Log (Word 3)	
Reg[1097] – Log	Valve Strokes Log (Word 1)	
Reg[1098] – Log	Valve Strokes Log (Word 2)	
Reg[1099] – Log	Valve Strokes Log (Word 3)	
Reg[1100] – Log Archive	Archive Motor Starts Log (Word 1)	

Reg[1101] – Log Archive	Archive Motor Starts Log (Word 2)	
Reg[1102] – Log Archive	Archive Motor Starts Log (Word 3)	
Reg[1103] – Log Archive	Archive Valve Strokes Log (Word 1)	
Reg[1104] – Log Archive	Archive Valve Strokes Log (Word 2)	
Reg[1105] – Log Archive	Archive Valve Strokes Log (Word 3)	
Reg[1106] – Log Archive	Archive Motor Run-time – Min	Archive Motor Run-Time – Sec
Reg[1107] – Log Archive	Archive Motor Run-time – Hours	
Reg[1108] – Real-Time Data	Current Valve Position (0.0% -100.0% in 0.1% increments)	
Reg[1109] – Real-Time Data	Current Valve Position (0 - 4095)	
Reg[1110] – Real-Time Data	Current Torque (0.0% -100.0% in 0.1% increments)	
Reg[1111] – Real-Time Data	Current Torque (0 - 4095)	
Reg[1112] – Real-Time Data	Analog Input #1 (0 - 4095)	
Reg[1113] – Real-Time Data	Analog Input #2 (0 - 4095)	
Reg[1114] – Real-Time Data	**Analog Output #1 (0 - 4095)**	
Reg[1115] – Real-Time Data	**Analog Output #2 (0 - 4095)**	
Reg[1116] – Real-Time Data	**Position Control Setpoint (0 - 4095)**	
Reg[1117] – Real-Time Data	**Position Control Setpoint (0.0% -100.0% in 0.1% increments)**	
Reg[1118] – User (Field) Setup	Analog Input #1 ZERO Calibration Data	
Reg[1119] – User (Field) Setup	Analog Input #1 SPAN Calibration Data	
Reg[1120] – User (Field) Setup	Analog Input #2 ZERO Calibration Data	
Reg[1121] – User (Field) Setup	Analog Input #2 SPAN Calibration Data	
Reg[1122] – User (Field) Setup	Analog Output #1 ZERO Calibration Data	
Reg[1123] – User (Field) Setup	Analog Output #1 SPAN Calibration Data	
Reg[1124] – User (Field) Setup	Analog Output #2 ZERO Calibration Data	
Reg[1125] – User (Field) Setup	Analog Output #2 SPAN Calibration Data	
Reg[1126] – Factory Setup	Valve Travel Speed (seconds if part-turn valve) or (RPM if multi-turn valve)	
Reg[1127] -	**(RESERVED – NO-OP)**	
Reg[1128] – ID	CCM Version ID – Mid Num	CCM Version ID – High Num
Reg[1129] – ID	CCM Compile Date – Char 0	CCM Version ID – Low Num
Reg[1130] – ID	CCM Compile Date – Char 2	CCM Compile Date – Char 1
Reg[1131] – ID	CCM Compile Date – Char 4	CCM Compile Date – Char 3
Reg[1132] – ID	CCM Compile Date – Char 6	CCM Compile Date – Char 5
Reg[1133] – ID	CCM Compile Date – Char 8	CCM Compile Date – Char 7
Reg[1134] – ID	**(RESERVED – NO-OP)**	
Reg[1135] – ID	**(RESERVED – NO-OP)**	
Reg[1136] – ID	RDM1 Version ID – Mid Num	RDM1 Version ID – High Num
Reg[1137] – ID	RDM1 Compile Date – Char 0	RDM1 Version ID – Low Num
Reg[1138] – ID	RDM1 Compile Date – Char 2	RDM1 Compile Date – Char 1
Reg[1139] – ID	RDM1 Compile Date – Char 4	RDM1 Compile Date – Char 3
Reg[1140] – ID	RDM1 Compile Date – Char 6	RDM1 Compile Date – Char 5
Reg[1141] – ID	RDM1 Compile Date – Char 8	RDM1 Compile Date – Char 7
Reg[1142] – ID	**(RESERVED – NO-OP)**	
Reg[1143] – ID	**(RESERVED – NO-OP)**	
Reg[1144] – ID	RDM2 Version ID – Mid Num	RDM2 Version ID – High Num
Reg[1145] – ID	RDM2 Compile Date – Char 0	RDM2 Version ID – Low Num
Reg[1146] – ID	RDM2 Compile Date – Char 2	RDM2 Compile Date – Char 1
Reg[1147] – ID	RDM2 Compile Date – Char 4	RDM2 Compile Date – Char 3
Reg[1148] – ID	RDM2 Compile Date – Char 6	RDM2 Compile Date – Char 5
Reg[1149] – ID	RDM2 Compile Date – Char 8	RDM2 Compile Date – Char 7
Reg[1150] – ID	**(RESERVED – NO-OP)**	
Reg[1151] – ID	**(RESERVED – NO-OP)**	
Reg[1152] – ID	LDM Version ID – Mid Num	LDM Version ID – High Num

Reg[1153] – ID	LDM Compile Date – Char 0	LDM Version ID – Low Num
Reg[1154] – ID	LDM Compile Date – Char 2	LDM Compile Date – Char 1
Reg[1155] – ID	LDM Compile Date – Char 4	LDM Compile Date – Char 3
Reg[1156] – ID	LDM Compile Date – Char 6	LDM Compile Date – Char 5
Reg[1157] – ID	LDM Compile Date – Char 8	LDM Compile Date – Char 7
Reg[1158] – ID	**(RESERVED – NO-OP)**	
Reg[1159] – ID	**(RESERVED – NO-OP)**	
Reg[1160] – ID	APD Version ID – Mid Num	APD Version ID – High Num
Reg[1161] – ID	APD Compile Date – Char 0	APD Version ID – Low Num
Reg[1162] – ID	APD Compile Date – Char 2	APD Compile Date – Char 1
Reg[1163] – ID	APD Compile Date – Char 4	APD Compile Date – Char 3
Reg[1164] – ID	APD Compile Date – Char 6	APD Compile Date – Char 5
Reg[1165] – ID	APD Compile Date – Char 8	APD Compile Date – Char 7
Reg[1166] – ID	**(RESERVED – NO-OP)**	
Reg[1167] – ID	**(RESERVED – NO-OP)**	
Reg[1168] – ID	ACM Version ID – Mid Num	ACM Version ID – High Num
Reg[1169] – ID	ACM Compile Date – Char 0	ACM Version ID – Low Num
Reg[1170] – ID	ACM Compile Date – Char 2	ACM Compile Date – Char 1
Reg[1171] – ID	ACM Compile Date – Char 4	ACM Compile Date – Char 3
Reg[1172] – ID	ACM Compile Date – Char 6	ACM Compile Date – Char 5
Reg[1173] – ID	ACM Compile Date – Char 8	ACM Compile Date – Char 7
Reg[1174] – ID	**(RESERVED – NO-OP)**	
Reg[1175] – ID	**(RESERVED – NO-OP)**	
Reg[1176] – Alarm Log	Alarm [1]	Alarm [0] … Newest
Reg[1177] – Alarm Log	Alarm [3]	Alarm [2]
Reg[1178] – Alarm Log	Alarm [5]	Alarm [4]
Reg[1179] – Alarm Log	Alarm [7]	Alarm [6]
Reg[1180] – Alarm Log	Alarm [9] … Oldest	Alarm [8]
Reg[1181] –	**Reserved – NO-OP**	**Reserved – NO-OP**
Reg[1182] –	**Reserved – NO-OP**	**Reserved – NO-OP**
Reg[1183] –	**Reserved – NO-OP**	**Reserved – NO-OP**
Reg[1184] – Event Log – Record (1)	Log Entry Number	
Reg[1185] – Event Log – Record (1)	Register [1000] … Inputs [1000 – 1015]	
Reg[1186] – Event Log – Record (1)	Register [1001] … Inputs [1016 – 1031]	
Reg[1187] – Event Log – Record (1)	Register [1002] … Inputs [1032 – 1047]	
Reg[1188] – Event Log – Record (1)	Time – minutes	Time – seconds
Reg[1189] – Event Log – Record (1)	Date – Day	Time – Hour
Reg[1190] – Event Log – Record (1)	Date – Year	Date – Month
Reg[1191] – Event Log – Record (1)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1192] – Event Log – Record (1)	CRC – High byte	CRC – Low Byte
Reg[1193] – Event Log – Record (2)	Log Entry Number	
Reg[1194] – Event Log – Record (2)	Register [1000] … Inputs [1000 – 1015]	
Reg[1195] – Event Log – Record (2)	Register [1001] … Inputs [1016 – 1031]	
Reg[1196] – Event Log – Record (2)	Register [1002] … Inputs [1032 – 1047]	
Reg[1197] – Event Log – Record (2)	Time - minutes	Time – seconds
Reg[1198] – Event Log – Record (2)	Date – Day	Time – Hour
Reg[1199] – Event Log – Record (2)	Date – Year	Date – Month
Reg[1200] – Event Log – Record (2)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1201] – Event Log – Record (2)	CRC – High byte	CRC – Low Byte
Reg[1202] – Event Log – Record (3)	Log Entry Number	
Reg[1203] – Event Log – Record (3)	Register [1000] … Inputs [1000 – 1015]	
Reg[1204] – Event Log – Record (3)	Register [1001] … Inputs [1016 – 1031]	
Reg[1205] – Event Log – Record (3)	Register [1002] … Inputs [1032 – 1047]	
Reg[1206] – Event Log – Record (3)	Time - minutes	Time – seconds

Reg[1207] – Event Log – Record (3)	Date – Day	Time – Hour
Reg[1208] – Event Log – Record (3)	Date – Year	Date – Month
Reg[1209] – Event Log – Record (3)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1210] – Event Log – Record (3)	CRC – High byte	CRC – Low Byte
Reg[1211] – Event Log – Record (4)	Log Entry Number	
Reg[1212] – Event Log – Record (4)	Register [1000] … Inputs [1000 – 1015]	
Reg[1213] – Event Log – Record (4)	Register [1001] … Inputs [1016 – 1031]	
Reg[1214] – Event Log – Record (4)	Register [1002] … Inputs [1032 – 1047]	
Reg[1215] – Event Log – Record (4)	Time - minutes	Time – seconds
Reg[1216] – Event Log – Record (4)	Date – Day	Time – Hour
Reg[1217] – Event Log – Record (4)	Date – Year	Date – Month
Reg[1218] – Event Log – Record (4)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1219] – Event Log – Record (4)	CRC – High byte	CRC – Low Byte
Reg[1220] – Event Log – Record (5)	Log Entry Number	
Reg[1221] – Event Log – Record (5)	Register [1000] … Inputs [1000 – 1015]	
Reg[1222] – Event Log – Record (5)	Register [1001] … Inputs [1016 – 1031]	
Reg[1223] – Event Log – Record (5)	Register [1002] … Inputs [1032 – 1047]	
Reg[1224] – Event Log – Record (5)	Time - minutes	Time – seconds
Reg[1225] – Event Log – Record (5)	Date – Day	Time – Hour
Reg[1226] – Event Log – Record (5)	Date – Year	Date – Month
Reg[1227] – Event Log – Record (5)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1228] – Event Log – Record (5)	CRC – High byte	CRC – Low Byte
Reg[1229] – Event Log – Record (6)	Log Entry Number	
Reg[1230] – Event Log – Record (6)	Register [1000] … Inputs [1000 – 1015]	
Reg[1231] – Event Log – Record (6)	Register [1001] … Inputs [1016 – 1031]	
Reg[1232] – Event Log – Record (6)	Register [1002] … Inputs [1032 – 1047]	
Reg[1233] – Event Log – Record (6)	Time - minutes	Time – seconds
Reg[1234] – Event Log – Record (6)	Date – Day	Time – Hour
Reg[1235] – Event Log – Record (6)	Date – Year	Date – Month
Reg[1236] – Event Log – Record (6)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1237] – Event Log – Record (6)	CRC – High byte	CRC – Low Byte
Reg[1238] – Event Log – Record (7)	Log Entry Number	
Reg[1239] – Event Log – Record (7)	Register [1000] … Inputs [1000 – 1015]	
Reg[1240] – Event Log – Record (7)	Register [1001] … Inputs [1016 – 1031]	
Reg[1241] – Event Log – Record (7)	Register [1002] … Inputs [1032 – 1047]	
Reg[1242] – Event Log – Record (7)	Time - minutes	Time – seconds
Reg[1243] – Event Log – Record (7)	Date – Day	Time – Hour
Reg[1244] – Event Log – Record (7)	Date – Year	Date – Month
Reg[1245] – Event Log – Record (7)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1246] – Event Log – Record (7)	CRC – High byte	CRC – Low Byte
Reg[1247] – Event Log – Record (8)	Log Entry Number	
Reg[1248] – Event Log – Record (8)	Register [1000] … Inputs [1000 – 1015]	
Reg[1249] – Event Log – Record (8)	Register [1001] … Inputs [1016 – 1031]	
Reg[1250] – Event Log – Record (8)	Register [1002] … Inputs [1032 – 1047]	
Reg[1251] – Event Log – Record (8)	Time - minutes	Time – seconds
Reg[1252] – Event Log – Record (8)	Date – Day	Time – Hour
Reg[1253] – Event Log – Record (8)	Date – Year	Date – Month
Reg[1254] – Event Log – Record (8)	Torque – (0 – 100%)	Position (0 – 100%)
Reg[1255] – Event Log – Record (8)	CRC – High byte	CRC – Low Byte
---------------------------------- END TEC2000 HOLDING REGISTERS ----------------------		

6.5. App Note: Modbus Message Formats

All Modbus communication timing estimates in this document are based on 9600 baud.

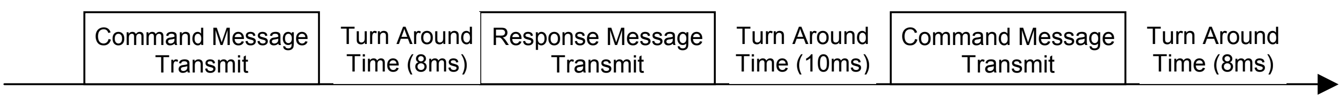
There is always a **10ms** minimum "Command Message Turn Around Time" between when the last message was received by the 1746-C and when it can transmit again … the physical network actually requires 8ms but 10ms is what the 1746-C delivers.

There is always a **8ms** minimum "Response Message Turn Around Time" between when the last message was received by the actuator and when it can transmit again.

At 9600 baud, it takes about **1.145ms / character** transmitted … per byte in the message being transmitted.

(11 bits in each char) / (9600 baud)

DIAGRAM OF COMMUNICATION ON THE NETWORK



6.5.1. Modbus RTU Functions (Command Codes) Implemented

1746-C system utilizes a subset of Modbus RTU protocol. The protocol is implemented in accordance with Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J.

The following Modbus function codes are used:

Code	Meaning	Action and Data Type
01	Read Coil Status	Read current status of a group of discrete output bits
03	Read Holding Register	Read binary value in one or more 16-bit registers
05	Set (Force) Single Coil	Write a single discrete output bit on or off
06	Set Single Register	Write a binary value to a specific 16-bit register
15	Set (Force) Multiple Coils	Write on/off state to a group of coils

6.5.2. Modbus Function (Command) Code Descriptions

Most of the rest of the information in this section came directly (or was paraphrased) from the Modbus Specification.

6.5.2.1. Modbus Function Code 01 … Read Coil Status

Using function code 01, the Modbus master (host) may directly address and read bits that indicate the live discrete outputs, software generated status bits and configuration bits.

Refer to the previous section that identifies "*Standard Coils*"

Example (in byte order) of a request to read 37 coils … 20–56 (internal coil addresses 19 – 55) from slave device 17:

1.	Slave Address	11
2.	Function	01
3.	Starting Address Hi byte	00
4.	Starting Address Lo byte	13
5.	No. of Coils Hi byte	00
6.	No. of Coils Lo byte	25
7.	CRC Hi byte	—
8.	CRC Lo byte	—

NOTE: 19 decimal is 0x13
37 decimal is 0x25

Response example (in byte order):

1.	Slave Address	11
2.	Function	01
3.	Byte Count	05
4.	Data (Coils 27–20)	CD
5.	Data (Coils 35–28)	6B
6.	Data (Coils 43–36)	B2
7.	Data (Coils 51–44)	0E
8.	Data (Coils 56–52)	1B
9.	CRC Hi byte	—
10.	CRC Lo byte	—

The status of coils 27–20 is shown as the byte value CD hex, or binary 1100 1101. Coil 27 is the MSB of this byte, and coil 20 is the LSB.
Left to right, the status of coils 27 through 20 is: ON–ON–OFF–OFF–ON–ON–OFF–ON.

By convention, bits within a byte are shown with the MSB to the left, and the LSB to the right. Thus the coils in the first byte are '27 through 20', from left to right.

The next byte has coils '35 through 28', left to right. As the bits are transmitted serially, they flow from LSB to MSB: 20 . . . 27, 28 . . . 35, and so on.

In the last data byte, the status of coils 56–52 is shown as the byte value 1B hex, or binary 0001 1011.
Coil 56 is in the fourth bit position from the left, and coil 52 is the LSB of this byte.
The status of coils 56 through 52 is: ON–ON–OFF–ON–ON.

Note how the three remaining bits (toward the high order end) are zero–filled.

6.5.2.2. Modbus Function Code 03 … Read Holding Register

Using function code 03, the Modbus master (host) may directly address and read any memory register in the actuator. This includes all registers containing discrete values (inputs & coils).

Refer to the previous section that describes the "*Addressable Holding Registers*".

Example (in byte order) of a request to read registers 40108 – 40110 (internal holding registers 107 – 109) from slave device 17:

1.	Slave Address	11
2.	Function	03
3.	Starting Address Hi byte	00
4.	Starting Address Lo byte	6B
5.	No. of Registers Hi byte	00
6.	No. of Registers Lo byte	03
7.	CRC Hi byte	—
8.	CRC Lo byte	—

NOTE: 107 decimal is 0x6B

Response example (in byte order):

1.	Slave Address	11
2.	Function	03
3.	Byte Count	06
4.	Data Hi (Register 40108)	02
5.	Data Lo (Register 40108)	2B
6.	Data Hi (Register 40109)	00
7.	Data Lo (Register 40109)	00
8.	Data Hi (Register 40110)	00
9.	Data Lo (Register 40110)	64
10.	CRC Hi byte	—
11.	CRC Lo byte	—

The contents of register 40108 are shown as the two byte values of 02 2B hex (or 555 decimal).

The contents of registers 40109–40110 are 00 00 and 00 64 hex, or 0 and 100 decimal.

6.5.2.3. Modbus Function Code 05 … Set (Force) Single Coil

Using function code 05, the Modbus master (host) may directly address and write single bits that indicate coils … to turn them ON or OFF.

Writing to a Read Only (RO) coil will
- Have no affect on valve operation
- Be ignored by the application software.
- Cause the actuator to return a response indicating "illegal function code".

If a configuration coil is set which is dependent upon other configuration coils being set (and the other coils are not set), a configuration error will be indicated that prevents valve actuator control.

When a valid configuration is written, the controller compares the state of the coil to the state stored in EEPROM. If the state has changed, the new state is stored to EEPROM configuration.

Refer to the previous section that identifies "*Standard Coils*"

Example (in byte order) of a request to force coil 173 (internal coil number 172) "ON" in slave device 17:

1.	Slave Address	11
2.	Function	05
3.	Coil Address Hi byte	00
4.	Coil Address Lo byte	AC
5.	Force Data Hi byte	FF
6.	Force Data Lo byte	00
7.	CRC Hi byte	—
8.	CRC Lo byte	—

NOTE: 172 decimal = 0xAC

Response example (in byte order):

1.	Slave Address	11
2.	Function	05
3.	Coil Address Hi	00
4.	Coil Address Lo	AC
5.	Force Data Hi	FF
6.	Force Data Lo	00
7.	CRC Hi byte	—
8.	CRC Lo byte	—

6.5.2.4. Modbus Function Code 06 … Set Single Register

Using function code 06, the Modbus master (host) may directly address and write any Read/Write (RW) memory register in the actuator. This includes registers that hold discrete values (inputs & coils).

Writing to a Read Only (RO) register will

- Have no affect on valve operation
- Be ignored by the application software.
- Cause the actuator to return a response indicating "illegal function code".

Writing to a configuration data register that is Read/Write (RW) will cause the controller to compare the written valve to the value stored in the EEPROM. If the value is different, the new value will be stored to EEPROM configuration.
If a configuration coil is set which is dependent upon other configuration coils being set (and the other coils are not set), a configuration error will be indicated that prevents valve actuator control.

Refer to the previous section that describes the "*Addressable Holding Registers*".

Example (in byte order) of a request to preset register 40002 (internal holding register 01) to 00 03 hex in slave device 17:

1.	Slave Address	11
2.	Function	06
3.	Register Address Hi byte	00
4.	Register Address Lo byte	01
5.	Preset Data Hi byte	00
6.	Preset Data Lo byte	03
7.	CRC Hi byte	—
8.	CRC Lo byte	—

Response example (in byte order):

1.	Slave Address	11
2.	Function	06
3.	Register Address Hi	00
4.	Register Address Lo	01
5.	Preset Data Hi	00
6.	Preset Data Lo	03
7.	CRC Hi byte	—
8.	CRC Lo byte	—

6.5.2.5. Modbus Function Code 15 (0x0F) … Set (Force) Multiple Coils

Using function code 15, the Modbus master (host) may directly address and write a range of consecutive bits that indicate coils … to turn them ON or OFF.

This is very similar to function code 05 that deals with single coils (register bits) except that function code 15 deals with a consecutive range of bits (coils).

For more information, refer to the previous section that identifies "*Function code 05 … Set (Force) Single Coil*"

Also, refer to the previous section that identifies "*Standard Coils*"

Example (in byte order) of a request to force a series of ten coils … starting at coil 20 (internal coil number 19 … 13 hex) in slave device 17.

1.	Slave Address		11
2.	Function		0F
3.	Coil Address Hi byte		00
4.	Coil Address Lo byte		13
5.	Quantity of Coils Hi byte		00
6.	Quantity of Coils Lo byte		0A
7.	Data Byte Count		02
8.	1st Data byte	(Coils 27-20)	CD
9.	2nd Data byte	(Coils 29-28)	01
10.	CRC Hi byte		—
11.	CRC Lo byte		—

The query data contents are two bytes: CD 01 hex
(1100 1101 0000 0001 binary).

The binary bits correspond to the coils in the following way:

```
Bit:   1  1  0  0  1  1  0  1    0 0 0 0 0 0  0  1
Coil: 27 26 25 24 23 22 21 20    - - - - - - 29 28
```

The first byte transmitted (CD hex) addresses coils 27-20, with the least significant bit addressing the lowest coil (20) in this set.

The next byte transmitted (01 hex) addresses coils 29-28, with the least significant bit addressing the lowest coil (28) in this set.

Unused bits in the last data byte should be zero–filled.

Response example (in byte order):

1.	Slave Address	11
2.	Function	0F
3.	Coil Address Hi	00
4.	Coil Address Lo	13
5.	Quantity of Coils Hi	00
6.	Quantity of Coils Lo	0A
7.	CRC Hi byte	—
8.	CRC Lo byte	—

6.5.3. Modbus Exception Messages Supported

There are 3 Modbus Exception Messages supported by the CPU (as responses back to the Modbus master) for Modbus function calls to these registers:

- Exception 01 →→ Illegal function code received. The query is not an allowable action for the slave.

- Exception 02 →→ Illegal data address received. The query is not an allowable address for the slave.

- Exception 03 →→ Illegal data value received. The value contained in the query field is not an allowable value for the slave.

This is a 5 byte packet whose structure is always the same such that

- Byte 1: The slave address responding
- Byte 2: The function code that generated the error (+ 0x80)
- Byte 3: The error exception type (01, 02 or 03)
- Byte 4: The CRC high byte
- Byte 5: The CRC low byte

For byte #2 (function code generating the error), the most significant bit of the original function code that generated the error/exception response is set to “1” to indicate which code had an error. For example:

- If function 1 (0x01) was sent to the slave and the slave had an exception with that function, it would return a 0x81 in byte #2.

- If function 16 (0x10) was sent to the slave and the slave had an exception with that function, it would return a 0x90 in byte #2.

Any “incorrectly formatted” message will return an Exception Code 02.

6.6. App Note: Install the 1746-C Network Master Firmware

Unless the 1746-C Network Master module is supplied by EIM with software (driver) installed, it is necessary to install it. Installation may also become necessary for software upgrades or in the case of a failed 1746-C module being replaced in the field.

Follow the procedures in this section to install the software. The software for the 1746-C is supplied on a Microsoft DOS compatible diskette in Intel hex format. If an Allen-Bradley compatible software package is not available for downloading the program, any terminal program with the ability to upload ASCII text files may be used (for instance … the terminal program, HyperTerminal, which is installed with Windows95/98/NT, can be used).

The firmware is stored in the EEPROM as discussed in the previous section. Depending upon the setup, the program is normally run from Random Access Memory (RAM) with Auto-start. Auto-start uses a small script to automatically begin executing the machine code in the EEPROM. This is useful when there has been a power outage to the PLC or the PLC requires maintenance.

6.6.1. Configure the Module for a Firmware Upload

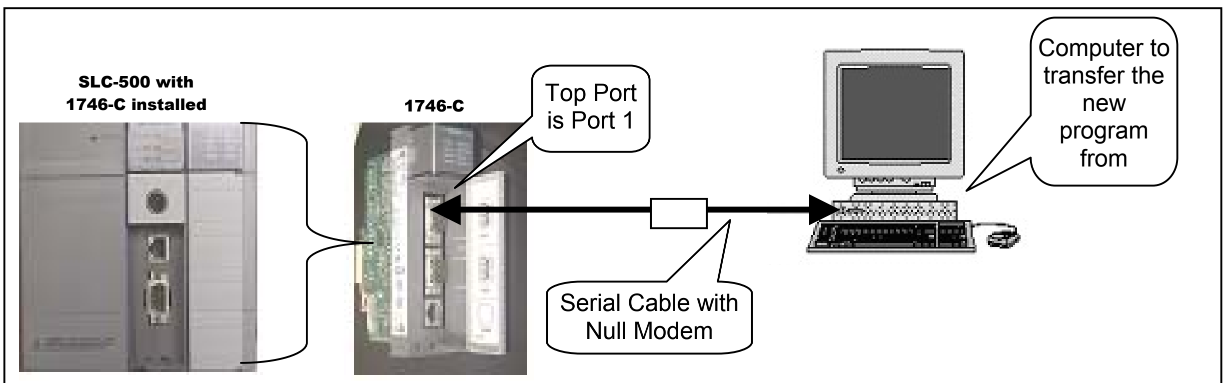
1. Disconnect the PRT1 and PRT2 connections to the 1746-C module.

2. Power down the PLC and pull the 1746-C module from the PLC rack.

3. Set JW-4 for
   - PRT1 = Default Program Mode (Default PGM).
   - PRT2 = ASCII.
   - DH485 = RUN.

4. Place the 1746-C module back in the rack.

5. Connect the serial communication cable between the 1746-C and the computer. The correct cable is a **Null Modem Cable** connected between the PRT1 port (top DB9 connector on the module) and the serial communications port on the computer used for the upload.



Sample Connection Diagram

6.6.2. Upload the 1746-C Firmware

Please refer to the Allen-Bradley SLC-500 BASIC User's Manual (1746-BAS & 1746-BAS-T)
Publication No. 1746-UM004A-US-P 2000

1. Make sure that the terminal program is running and that it is set for 1200 Baud, 8,N,1 and Software Flow Control (XON/XOFF).

2. Power the PLC rack up.

3. While in the terminal program, the startup screen for the 1746-C will be transmitted on power up. This also occurs on module reset as well.

4. Press <ENTER> several times to make sure that there is correct communication.
   (*The ">", a line item caret, will appear every time you press it.*)

> If you want (and can) communicate at a 9600 baud rather than 1200 baud, perform the following NOW:
> 1. Type in the command : PUSH 9600
> 2. Type in the command: CALL 78
> 3. Change your terminal program over to 9600 baud.
> 4. Continue with the standard procedure.

5. Type in the command "CALL 100".
   (*The module then waits for an upload from the terminal program.*)

------- ---------

6. Use the "Send Text File" command in your particular terminal program to send the file.
   The appropriate file is in "Intel hex format" (*which is an ASCII text file*) … all other formats will create an error.

------- ---------

7. After the file is finished being transmitted, the ">" will appear on the terminal screen to indicate finished.

8. Type "PROG2" and press <ENTER>.
   The module will respond back with a successful dialog. If it doesn't, repeat the previous step.

6.6.3. Reset the 1746-C Module for Normal Operations

1. Power down the PLC.

2. Pull the 1746-C module from the PLC rack.

3. Set JW-4 back to normal …
   - PRT1 = ASCII.
   - PRT2 = ASCII.
   - DH485 = PGM.

4. Put the 1746-C module back into the desired slot.

5. Reconnect the PRT1 and PRT2 connections to the 1746-C Network Master.

6. Power the module up with the rack.
   (*LED1 & LED2 should light up and PRT1 & PRT2 should flicker*)

7. Finished.